Codes from Yi Li, MQE, UCLA

In [1]:
```
import requests
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.formula.api as smf
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import math
from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.seasonal import seasonal_decompose
import warnings
warnings.filterwarnings("ignore")
```

I. Introduction

Our dataset contains 4-year samples (total 1461) of Seattle daily weather conditions from 2012-01-01 to 2015-12-31, columns including precipitation, maxmium temprature, minimal temprature, average temprature, wind, and description of weather conditions, all daily-based. To better understand the dataset, we transform the unit of temprature data from Degree Celsius to Fahrenheit scale. We include data descreptive statistic of qualitative variables below, and make sure there are no null values to further our exploration. For the main analysis of modeling trend, we choose to focus on the time-series data of average temprature in the dataset, and its density plot is included below. Within the average temprature, its mean is 54.2, standard deviation is 10.8, and range is around 55, from the histogram we could see that the distribution is not normal. Our goal of the project is to make 1-year-ahead prediction on the average temprature in Seattle.

In [2]:
```
df = pd.read_csv("seattle-weather.csv", index_col = 0, parse_dates = True)
df['temp_avg'] = (df['temp_avg'] * 9/5) + 32
df.head()
```

Out[2]:

date	precipitation	temp_max	temp_min	temp_avg	wind	weather
2012-01-01	0.0	12.8	5.0	48.02	4.7	drizzle
2012-01-02	10.9	10.6	2.8	44.06	4.5	rain
2012-01-03	0.8	11.7	7.2	49.01	2.3	rain
2012-01-04	20.3	12.2	5.6	48.02	4.7	rain
2012-01-05	1.3	8.9	2.8	42.53	6.1	rain

In [3]:
```
df.describe()
```

Out[3]:

	precipitation	temp_max	temp_min	temp_avg	wind
count	1461.000000	1461.000000	1461.000000	1461.000000	1461.000000
mean	3.029432	16.439083	8.234771	54.206468	3.241136
std	6.680194	7.349758	5.023004	10.796492	1.437825
min	0.000000	-1.600000	-7.100000	25.160000	0.400000
25%	0.000000	10.600000	4.400000	45.950000	2.200000
50%	0.000000	15.600000	8.300000	53.510000	3.000000
75%	2.800000	22.200000	12.200000	62.960000	4.000000
max	55.900000	35.600000	18.300000	80.060000	9.500000

In [4]:
```
df.isnull().sum()
```

Out[4]:
```
precipitation    0
temp_max         0
temp_min         0
temp_avg         0
wind             0
weather          0
dtype: int64
```

In [5]:
```
df.index
```

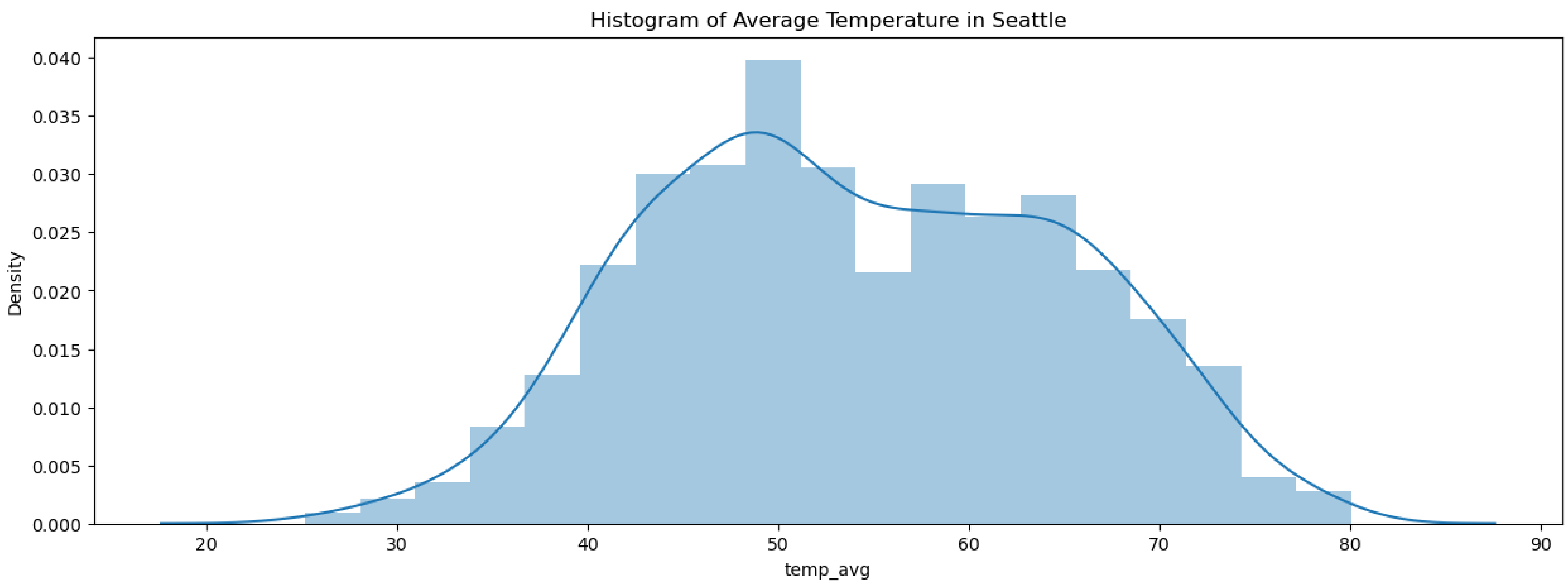
Out[5]:
```
DatetimeIndex(['2012-01-01', '2012-01-02', '2012-01-03', '2012-01-04',
               '2012-01-05', '2012-01-06', '2012-01-07', '2012-01-08',
               '2012-01-09', '2012-01-10',
               ...
               '2015-12-22', '2015-12-23', '2015-12-24', '2015-12-25',
               '2015-12-26', '2015-12-27', '2015-12-28', '2015-12-29',
               '2015-12-30', '2015-12-31'],
              dtype='datetime64[ns]', name='date', length=1461, freq=None)
```

In [6]:
```
df.index.nunique()
```

Out[6]:
```
1461
```

In [46]:
```
plt.figure(figsize = (15, 5))
sns.distplot(df["temp_avg"])
plt.title("Histogram of Average Temperature in Seattle")
```

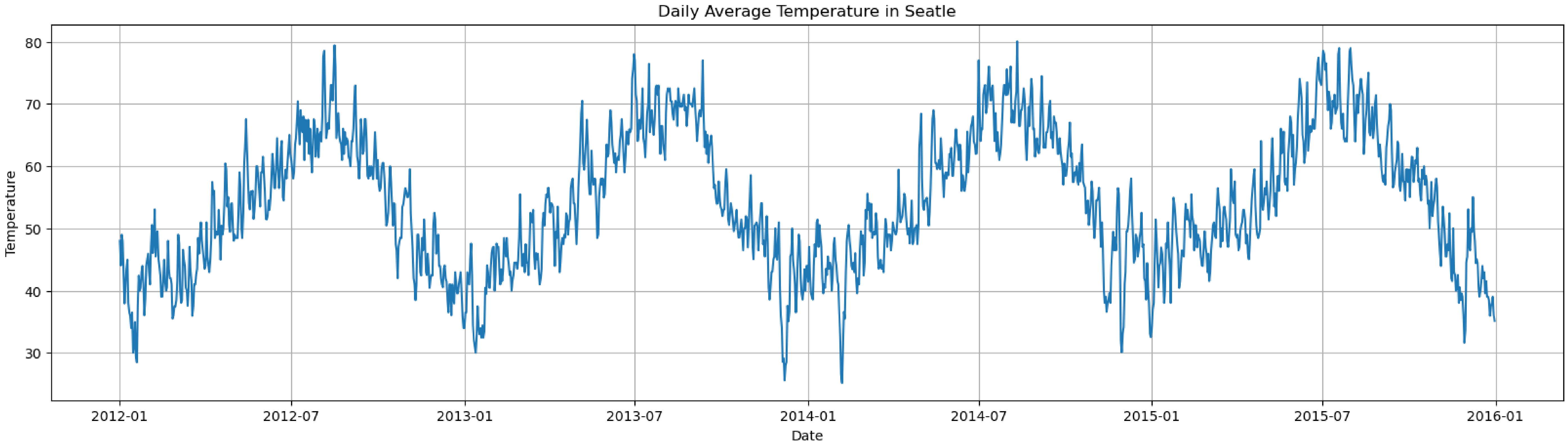
Out[46]:
```
Text(0.5, 1.0, 'Histogram of Average Temperature in Seattle')
```



II. Results

1. Modeling and Forecasting Trend

In [8]:
```
plt.figure(figsize = (20, 5))
plt.title('Daily Average Temperature in Seatle')
plt.plot(df["temp_avg"])
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.grid()
```

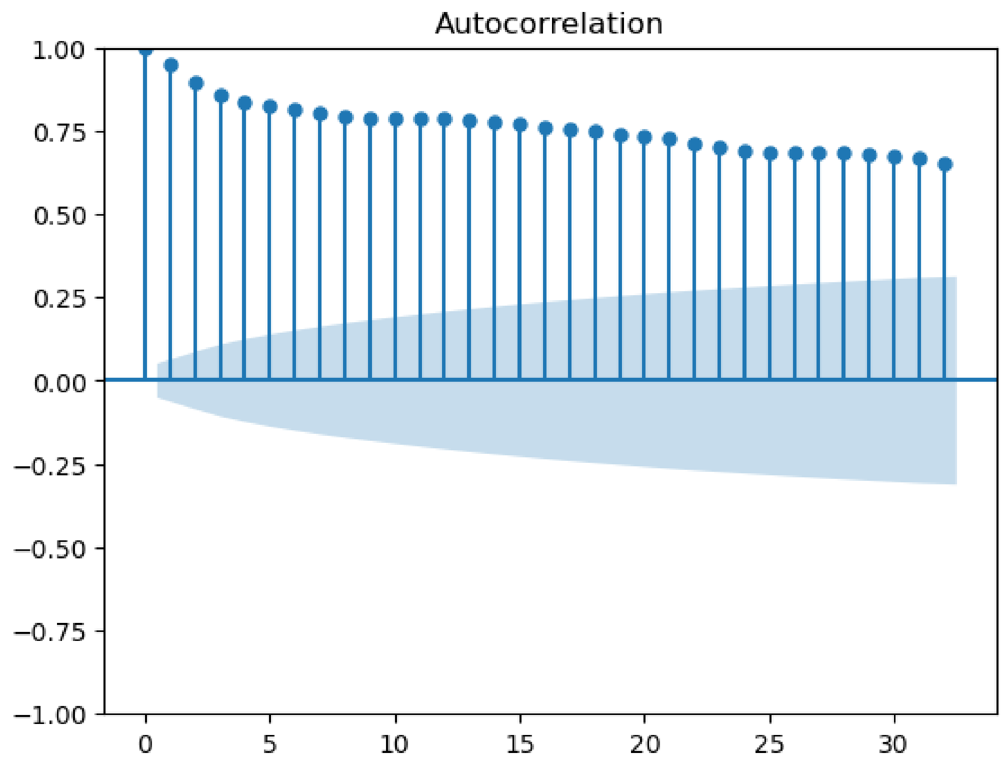


In [9]:
```
adf = adfuller(df["temp_avg"])[1]
print(f"p value:{adf}", ", Series is Stationary" if adf <0.05 else ", Series is Non-Stationary")
```

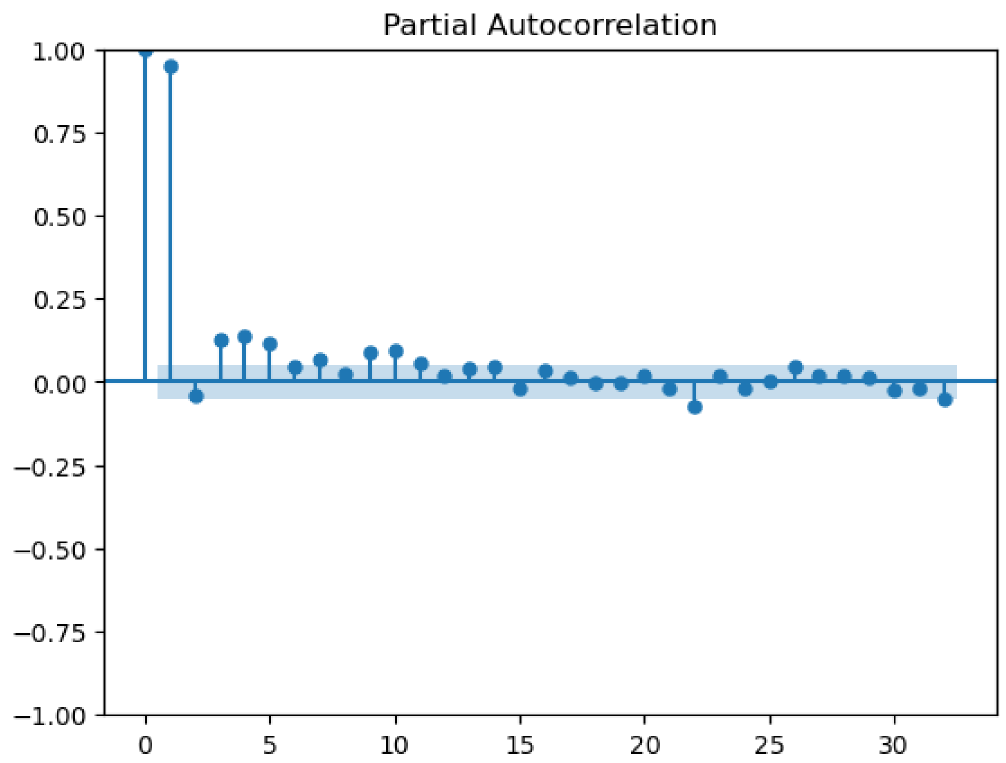
```
p value:0.169785824508232 , Series is Non-Stationary
```

The plot in (a) does not suggest the data are covariance stationary, because we could see that the mean value of the series data is time-varing. Moreover, we can tell that there is clear pattern exist in the series data, suggesting deterministic cycle and seasonality. Lastly, we perform the Augmented Dickey-Fuller test and obtain the resulting p-value being 0.17, which is greater than significance of 0.05, implying that the series is not stationary

In [10]:
```
plot_acf(df["temp_avg"]);
```



In [11]:
```
plot_pacf(df["temp_avg"], method = "ywm");
```



From the acf plot above, all lags are statistically significant and decreasing with slow pace. In the pacf plot, there are also several statistically significant of autocorrelation between lags such as 1, 3, 4, 5, 7, 9, 10.

We find quadratic + periodic (nonliinear) model fits our data the best from the regressions and respective plot of fit below.

Model1: Linear

In [12]:
```
df["t"] = np.arange(1, len(df)+1)
model1 = smf.ols("temp_avg ~ t", df).fit()
model1.summary()
```

Out[12]:

OLS Regression Results

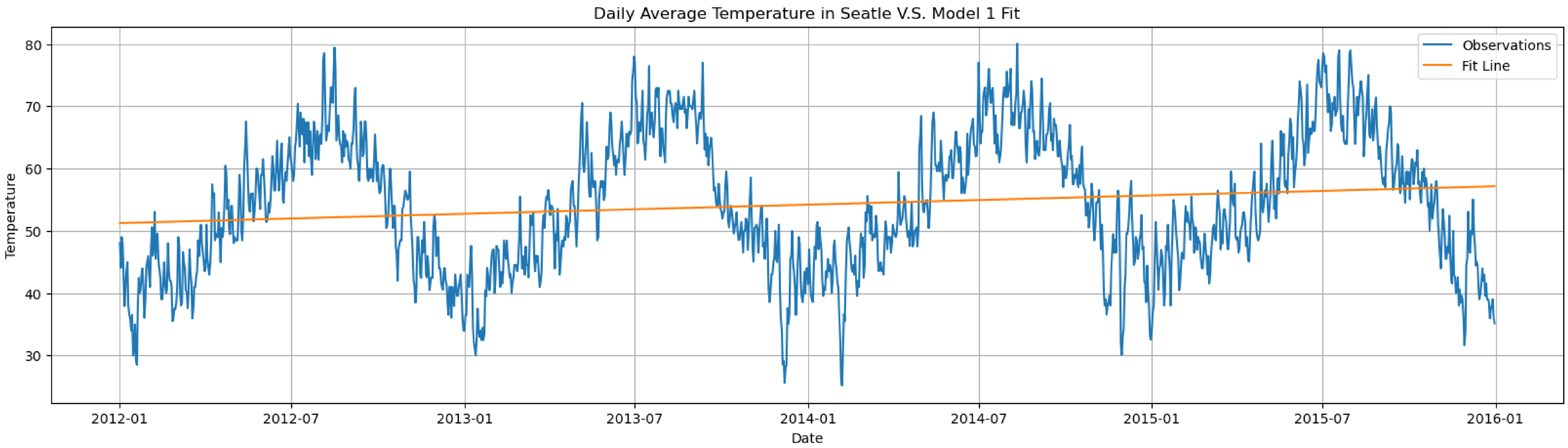
Dep. Variable:	temp_avg	**R-squared:**	0.025
Model:	OLS	**Adj. R-squared:**	0.024
Method:	Least Squares	**F-statistic:**	37.41
Date:	Sat, 22 Apr 2023	**Prob (F-statistic):**	1.23e-09
Time:	17:10:33	**Log-Likelihood:**	-5530.1
No. Observations:	1461	**AIC:**	1.106e+04
Df Residuals:	1459	**BIC:**	1.107e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	51.2486	0.558	91.795	0.000	50.153	52.344
t	0.0040	0.001	6.117	0.000	0.003	0.005

Omnibus:	84.032	**Durbin-Watson:**	0.105
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	32.250
Skew:	0.043	**Prob(JB):**	9.93e-08
Kurtosis:	2.277	**Cond. No.**	1.69e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.69e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

In [13]:
```
df["y_hat_m1"] = model1.fittedvalues
plt.figure(figsize = (20, 5))
plt.title('Daily Average Temperature in Seatle V.S. Model 1 Fit')
plt.plot(df["temp_avg"])
plt.plot(df["y_hat_m1"])
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.legend(["Observations", "Fit Line"])
plt.grid()
```



Model 2: Nonlinear (polynomial)

In [14]:
```
model2 = smf.ols("temp_avg ~ t + I(t**2)", df).fit()
model2.summary()
```

Out[14]:

OLS Regression Results

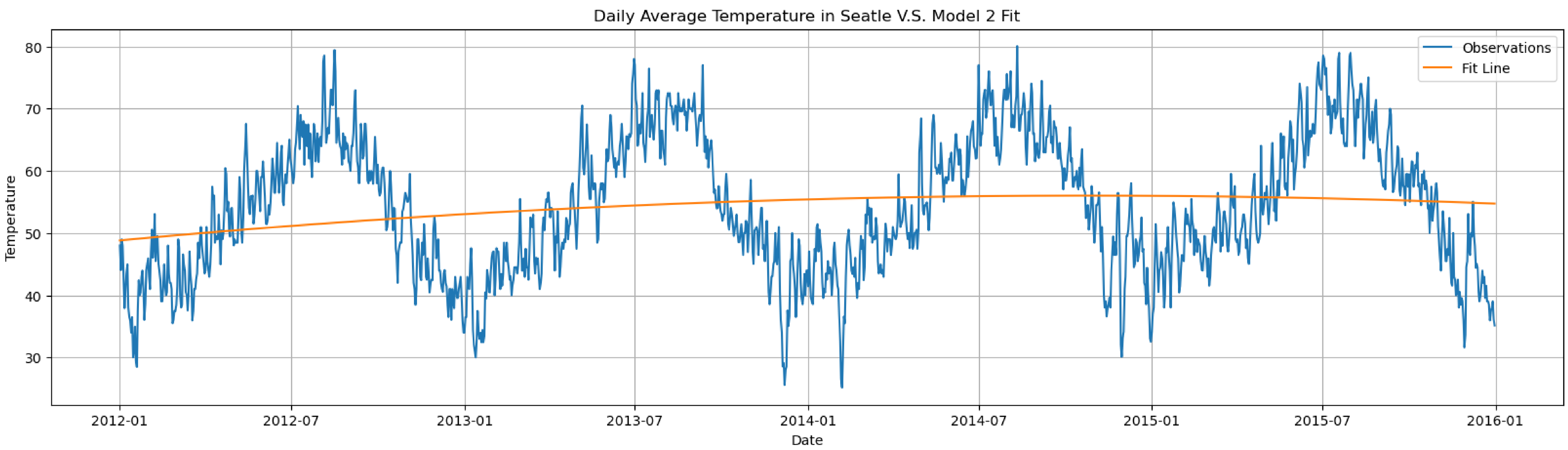
Dep. Variable:	temp_avg	**R-squared:**	0.035
Model:	OLS	**Adj. R-squared:**	0.034
Method:	Least Squares	**F-statistic:**	26.47
Date:	Sat, 22 Apr 2023	**Prob (F-statistic):**	5.10e-12
Time:	17:10:39	**Log-Likelihood:**	-5522.6
No. Observations:	1461	**AIC:**	1.105e+04
Df Residuals:	1458	**BIC:**	1.107e+04
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	48.8258	0.834	58.536	0.000	47.190	50.462
t	0.0140	0.003	5.306	0.000	0.009	0.019
I(t ** 2)	-6.796e-06	1.75e-06	-3.894	0.000	-1.02e-05	-3.37e-06

Omnibus:	75.699	**Durbin-Watson:**	0.106
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	30.240
Skew:	0.036	**Prob(JB):**	2.71e-07
Kurtosis:	2.299	**Cond. No.**	2.87e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.87e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [15]:
```
df["y_hat_m2"] = model2.fittedvalues
plt.figure(figsize = (20, 5))
plt.title('Daily Average Temperature in Seatle V.S. Model 2 Fit')
plt.plot(df["temp_avg"])
plt.plot(df["y_hat_m2"])
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.legend(["Observations", "Fit Line"])
plt.grid()
```



Model 3: Nonlinear (Quadratic + Periodic)

In [16]:
```
df["sint"] = [math.sin((2*math.pi/365)*i) for i in df["t"]]
df["cost"] = [math.cos((2*math.pi/365)*i) for i in df["t"]]
model3 = smf.ols("temp_avg ~ t + I(t**2) + sint + cost", df).fit()
model3.summary()
```

Out[16]: OLS Regression Results

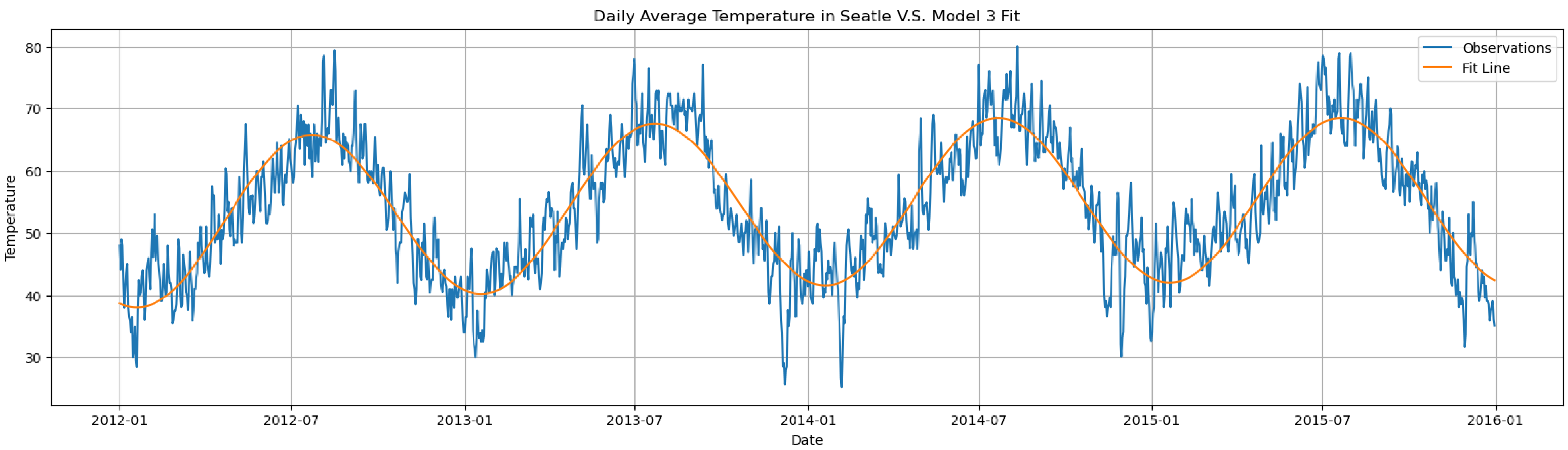
Dep. Variable:	temp_avg	**R-squared:**	0.787
Model:	OLS	**Adj. R-squared:**	0.786
Method:	Least Squares	**F-statistic:**	1345.
Date:	Sat, 22 Apr 2023	**Prob (F-statistic):**	0.00
Time:	17:10:46	**Log-Likelihood:**	-4419.1
No. Observations:	1461	**AIC:**	8848.
Df Residuals:	1456	**BIC:**	8875.
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	51.1331	0.395	129.390	0.000	50.358	51.908
t	0.0075	0.001	6.014	0.000	0.005	0.010
I(t ** 2)	-3.344e-06	8.22e-07	-4.067	0.000	-4.96e-06	-1.73e-06
sint	-4.6913	0.188	-24.915	0.000	-5.061	-4.322
cost	-12.4271	0.185	-67.201	0.000	-12.790	-12.064

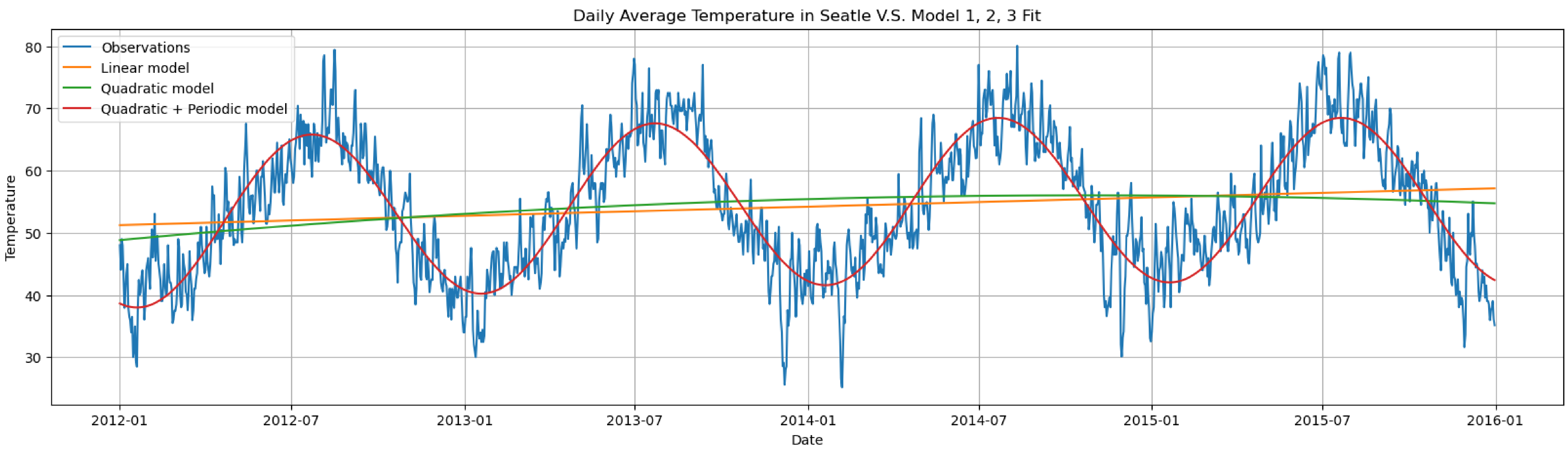
Omnibus:	5.703	**Durbin-Watson:**	0.480
Prob(Omnibus):	0.058	**Jarque-Bera (JB):**	6.184
Skew:	-0.090	**Prob(JB):**	0.0454
Kurtosis:	3.263	**Cond. No.**	2.90e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.9e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [17]:
```
df["y_hat_m3"] = model3.fittedvalues
plt.figure(figsize = (20, 5))
plt.title('Daily Average Temperature in Seatle V.S. Model 3 Fit')
plt.plot(df["temp_avg"])
plt.plot(df["y_hat_m3"])
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.legend(["Observations", "Fit Line"])
plt.grid()
```



In [18]:
```
plt.figure(figsize = (20, 5))
plt.title('Daily Average Temperature in Seatle V.S. Model 1, 2, 3 Fit')
plt.plot(df["temp_avg"])
plt.plot(df["y_hat_m1"])
plt.plot(df["y_hat_m2"])
plt.plot(df["y_hat_m3"])
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.legend(["Observations", "Linear model", "Quadratic model", "Quadratic + Periodic model"])
plt.grid()
```

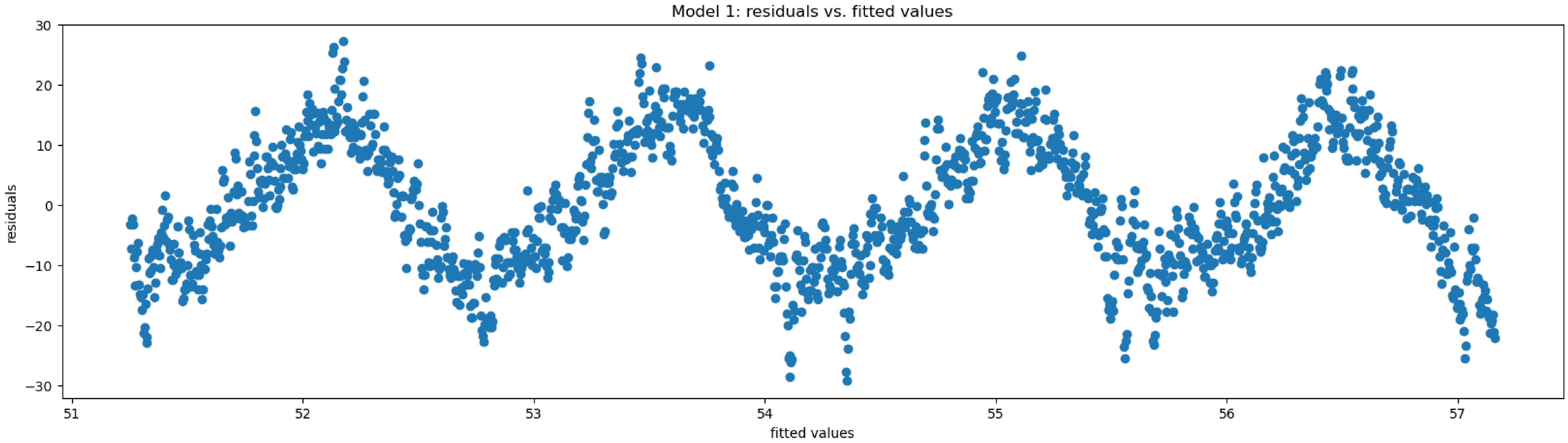


We could see that for model 3, the residuals are randomly distributing around 0 and that vairance is constant to imply homoskedasticity. But for linear and quadratic model, the residuals are not randomly scattering around 0 and suggesting non-constant variance of heteroskedasticity. Besides, there is a clear pattern of residuals for both model 1 and model 2.

Model 1

In [19]:
```
plt.figure(figsize = (20, 5))
plt.scatter(model1.fittedvalues, model1.resid)
plt.title('Model 1: residuals vs. fitted values')
plt.xlabel("fitted values")
plt.ylabel("residuals")
```

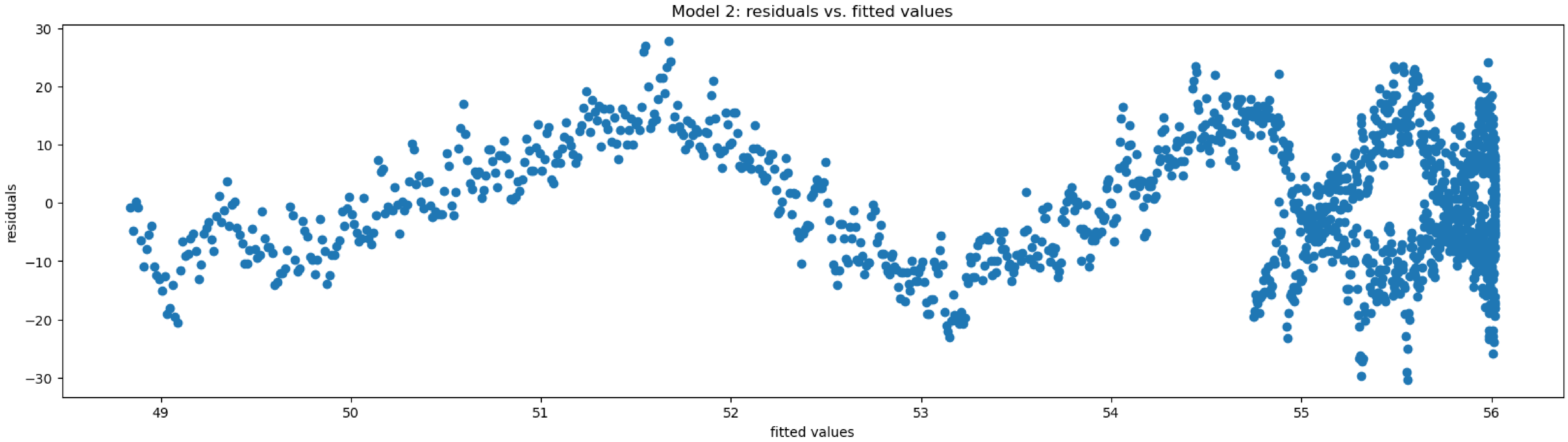
Out[19]: Text(0, 0.5, 'residuals')



Model 2

In [20]:
```
plt.figure(figsize = (20, 5))
plt.scatter(model2.fittedvalues, model2.resid)
plt.title('Model 2: residuals vs. fitted values')
plt.xlabel("fitted values")
plt.ylabel("residuals")
```

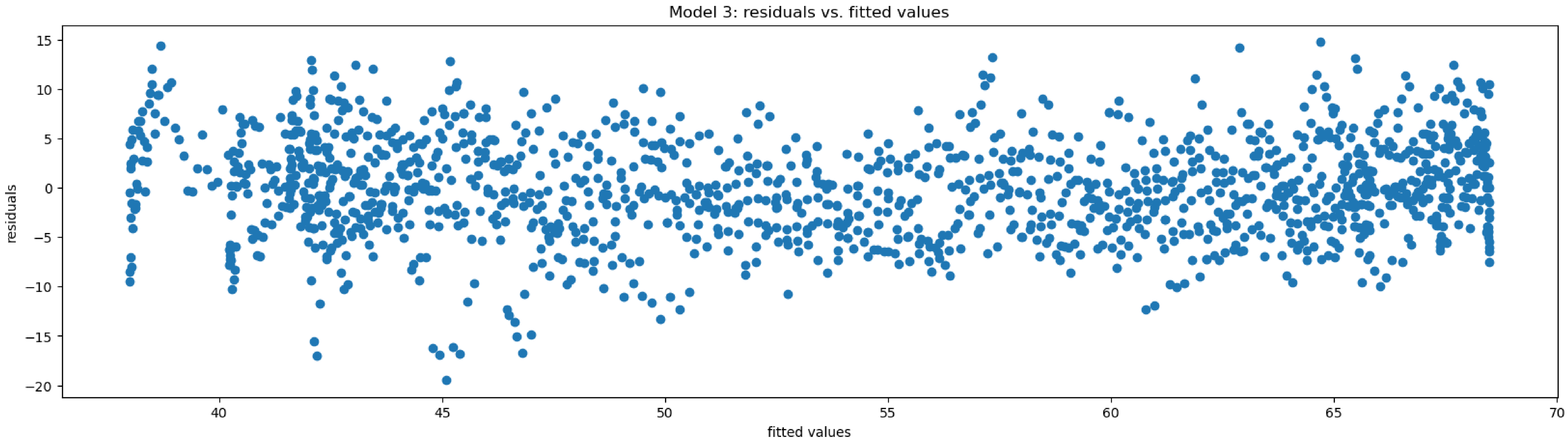
Out[20]: Text(0, 0.5, 'residuals')



Model 3

In [21]:
```
plt.figure(figsize = (20, 5))
plt.scatter(model3.fittedvalues, model3.resid)
plt.title('Model 3: residuals vs. fitted values')
plt.xlabel("fitted values")
plt.ylabel("residuals")
```

Out[21]: Text(0, 0.5, 'residuals')

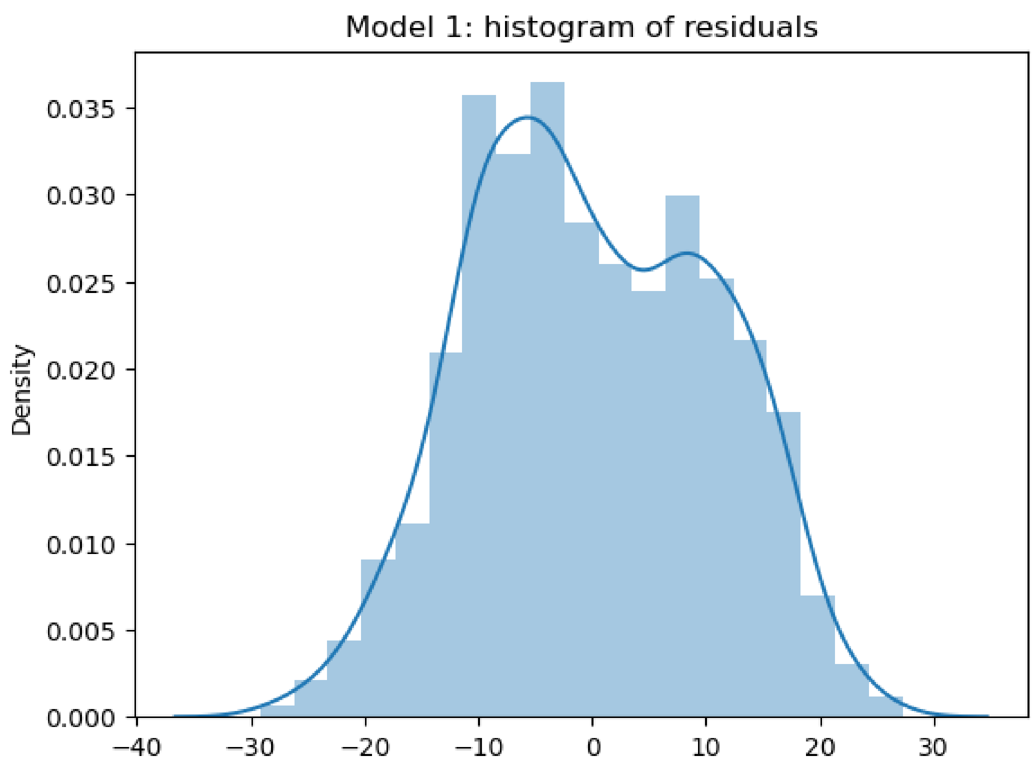


The range of residuals is the smallest for quadratic + period model (model 3), and almost normally distributed. The ranges of other two models are almost 20 larger than that of model 3 and obviously not normally distributed from the histograms of each residual of the regression below.

Model 1

In [22]:
```
sns.distplot(model1.resid)
plt.title('Model 1: histogram of residuals')
```

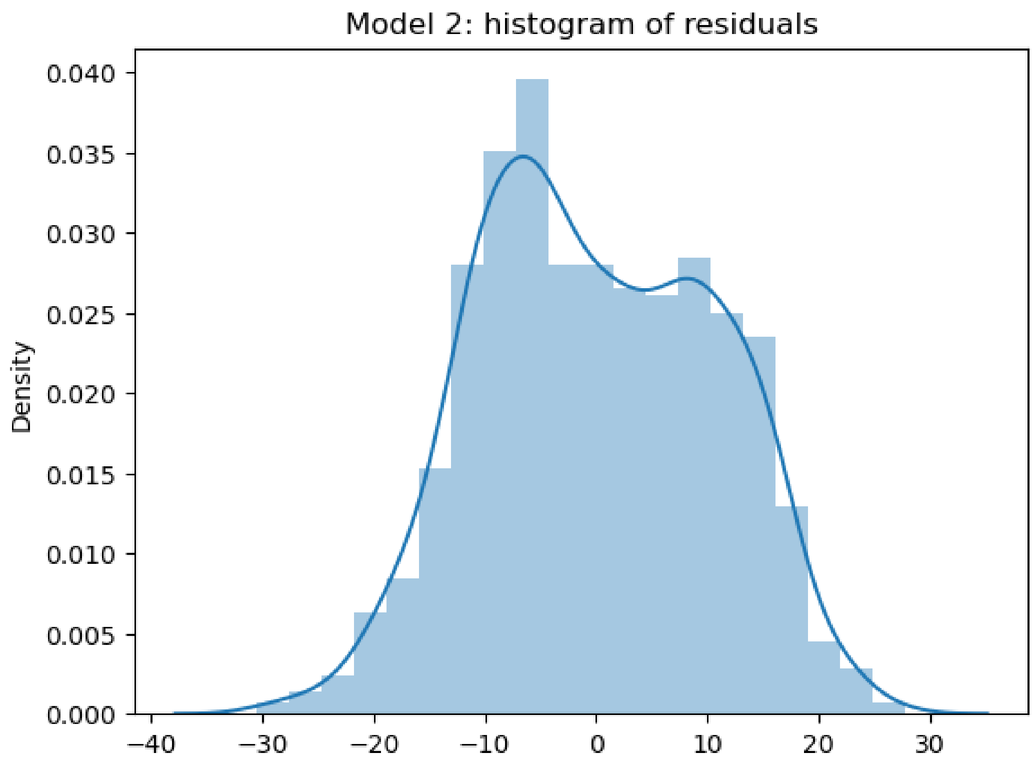
Out[22]: Text(0.5, 1.0, 'Model 1: histogram of residuals')



Model 2

In [23]:
```
sns.distplot(model2.resid)
plt.title('Model 2: histogram of residuals')
```

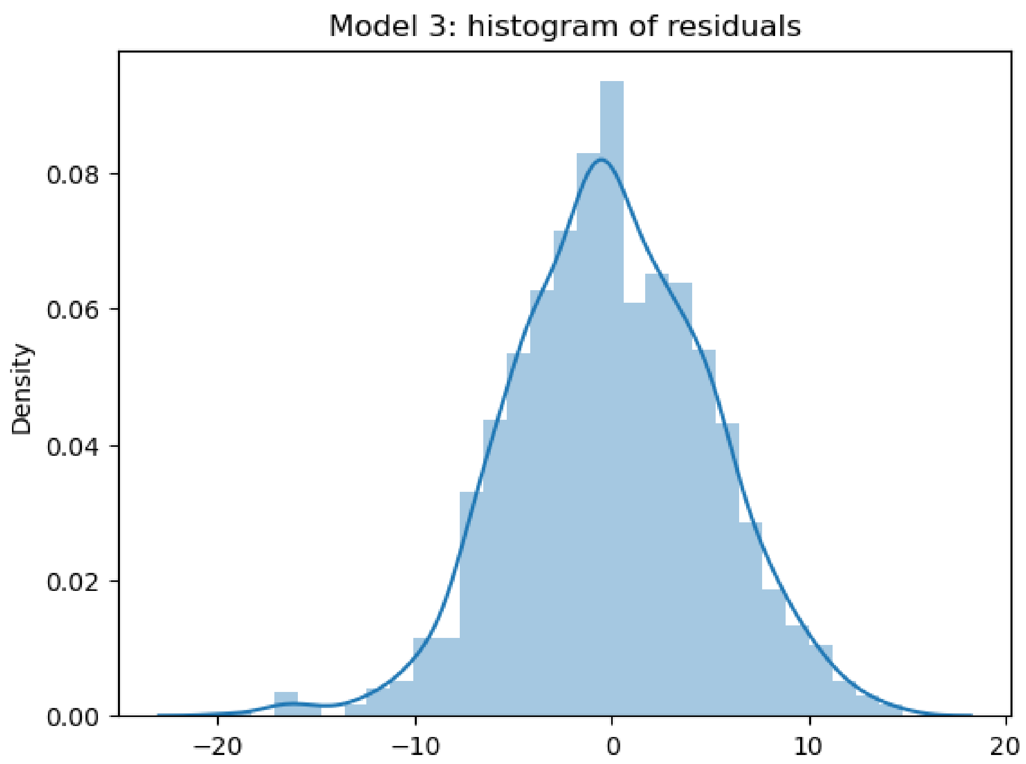
Out[23]:
```
Text(0.5, 1.0, 'Model 2: histogram of residuals')
```



Model 3

In [24]:
```
sns.distplot(model3.resid)
plt.title('Model 3: histogram of residuals')
```

Out[24]:
```
Text(0.5, 1.0, 'Model 3: histogram of residuals')
```



From the table below, we could see that the coefficients of all regressions are statistically significant. Noticably, the statistics of R-squared and Adjusted R-squared are the highest for model 3, and that the p_value of F-STAT suggests that the coefficients of model 3 regression are strongly jointly significant.

In [25]:
```
result = pd.DataFrame(np.ones((3,4)), columns = ["r_squared", "adjr_squared", "F_stat", "F_pvalue"])
result.index = ["Model 1", "Model 2", "Model 3"]
result.iloc[0, 0] = model1.rsquared
result.iloc[1, 0] = model2.rsquared
result.iloc[2, 0] = model3.rsquared
result.iloc[0, 1] = model1.rsquared_adj
result.iloc[1, 1] = model2.rsquared_adj
result.iloc[2, 1] = model3.rsquared_adj
result.iloc[0, 2] = model1.fvalue
result.iloc[1, 2] = model2.fvalue
result.iloc[2, 2] = model3.fvalue
result.iloc[0, 3] = model1.f_pvalue
result.iloc[1, 3] = model2.f_pvalue
result.iloc[2, 3] = model3.f_pvalue
```

In [26]:
```
result
```

Out[26]:

	r_squared	adjr_squared	F_stat	F_pvalue
Model 1	0.025002	0.024334	37.413122	1.225199e-09
Model 2	0.035039	0.033715	26.471070	5.099930e-12
Model 3	0.786955	0.786369	1344.557113	0.000000e+00

In [27]:
```
result_m1 = pd.DataFrame({"t_stat": model1.tvalues, "t_pvalue":model1.pvalues})
result_m1
```

Out[27]:

	t_stat	t_pvalue
Intercept	91.795379	0.000000e+00
t	6.116627	1.225199e-09

In [28]:
```
result_m2 = pd.DataFrame({"t_stat": model2.tvalues, "t_pvalue":model2.pvalues})
result_m2
```

Out[28]:

	t_stat	t_pvalue
Intercept	58.536022	0.000000e+00
t	5.306401	1.290845e-07
I(t ** 2)	-3.894325	1.029258e-04

In [29]:
```
result_m3 = pd.DataFrame({"t_stat": model3.tvalues, "t_pvalue":model3.pvalues})
result_m3
```

Out[29]:

	t_stat	t_pvalue
Intercept	129.389561	0.000000e+00
t	6.014126	2.283392e-09
I(t ** 2)	-4.067130	5.014430e-05
sint	-24.914538	2.041742e-114
cost	-67.200764	0.000000e+00

AIC and BIC agree on the model selection for the trend model with both being smallest (8848.15 and 8874.58 respectively) for quadratic + periodic model (model 3).

In [30]:
```
result = pd.DataFrame(np.ones((3,2)), columns = ["AIC", "BIC"])
result.index = ["Model 1", "Model 2", "Model 3"]
result.iloc[0, 0] = model1.aic
result.iloc[1, 0] = model2.aic
result.iloc[2, 0] = model3.aic
result.iloc[0, 1] = model1.bic
result.iloc[1, 1] = model2.bic
result.iloc[2, 1] = model3.bic
result
```

Out[30]:

	AIC	BIC
Model 1	11064.230597	11074.804350
Model 2	11051.112120	11066.972749
Model 3	8848.150538	8874.584920

We utilize quadratic + periodic model to make 1-year-ahead prediction (365-day-ahead on daily basis), from 2016-01-01 to 2016-12-30. The table below is statistic summary of 1-year-ahead prediction, including mean, standard error, 95% confidence interval, and 95% prediction interval. The plot below is the prediction following the original series data, including both 95% confidence interval of the prediction and 95% prediction interval of the prediction.

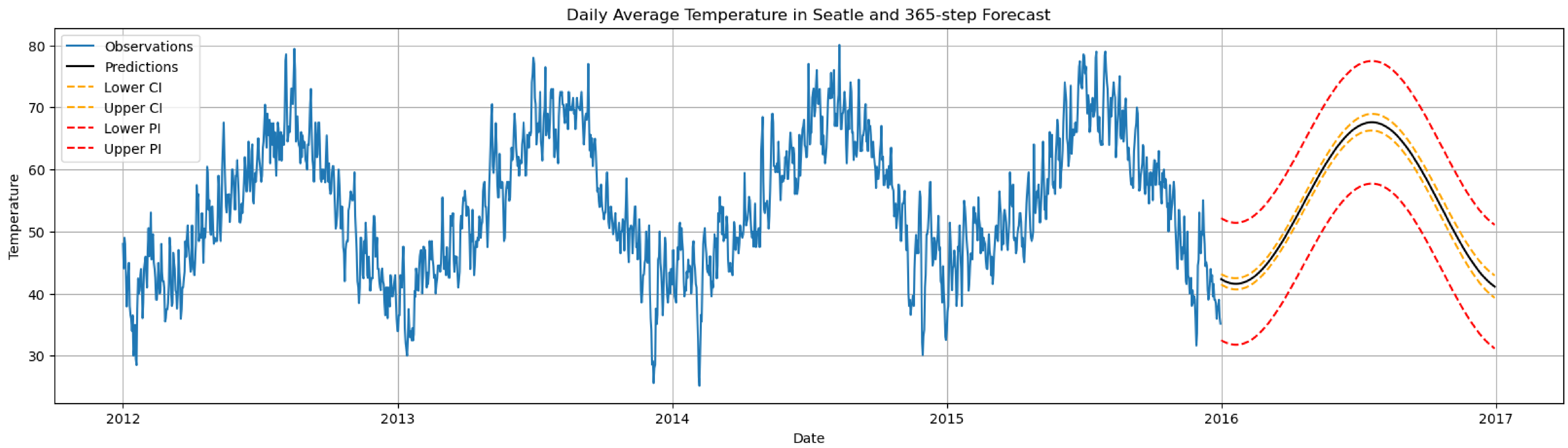
In [31]:
```
pred_df = pd.DataFrame()
t_pred = np.arange(len(df)+1, len(df)+366)
date_pred = pd.date_range(start = '01/01/2016', periods = 365, freq = 'd')
pred_df["t"] = t_pred
pred_df["sint"] = [math.sin((2*math.pi/365)*i) for i in pred_df["t"]]
pred_df["cost"] = [math.cos((2*math.pi/365)*i) for i in pred_df["t"]]
pred_df.index = date_pred
predictions = model3.get_prediction(pred_df)
predictions = predictions.summary_frame(alpha = 0.05)
predictions.index = date_pred
predictions
```

Out[31]:

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
2016-01-01	42.331600	0.428818	41.490433	43.172767	32.506856	52.156344
2016-01-02	42.257815	0.430123	41.414088	43.101541	32.432851	52.082778
2016-01-03	42.187772	0.431432	41.341477	43.034067	32.362587	52.012956
2016-01-04	42.121492	0.432746	41.272619	42.970365	32.296085	51.946899
2016-01-05	42.058993	0.434065	41.207534	42.910452	32.233363	51.884624
...	...	...	...	...	...	...
2016-12-26	41.481846	0.917497	39.682089	43.281604	31.529100	51.434593
2016-12-27	41.387247	0.919445	39.583669	43.190824	31.433809	51.340684
2016-12-28	41.296273	0.921397	39.488864	43.103681	31.342140	51.250405
2016-12-29	41.208950	0.923355	39.397702	43.020198	31.254119	51.163781
2016-12-30	41.125303	0.925317	39.310205	42.940401	31.169771	51.080835

365 rows × 6 columns

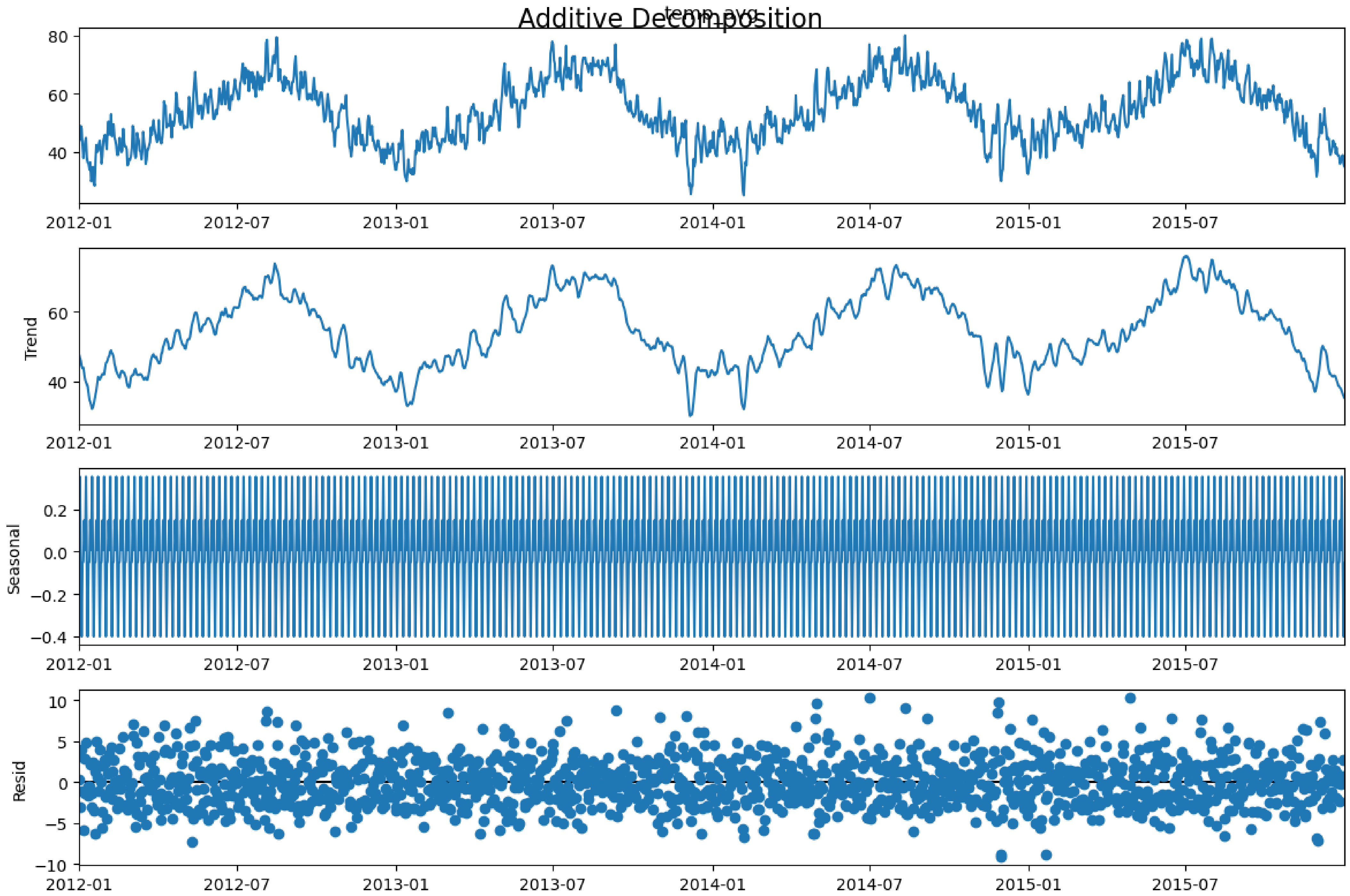
In [32]:
```
plt.figure(figsize = (20, 5))
plt.plot(df["temp_avg"])
plt.plot(predictions["mean"], color = "black")
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.title("Daily Average Temperature in Seatle and 365-step Forecast")

plt.plot(predictions["mean_ci_lower"], color = "orange", linestyle = "--")
plt.plot(predictions["mean_ci_upper"], color = "orange", linestyle = "--")

plt.plot(predictions["obs_ci_lower"], color = "red", linestyle = "--")
plt.plot(predictions["obs_ci_upper"], color = "red", linestyle = "--")

plt.legend(["Observations", "Predictions", "Lower CI", "Upper CI", "Lower PI", "Upper PI"])
plt.grid()
```



2. Trend and Seasonal Adjustments

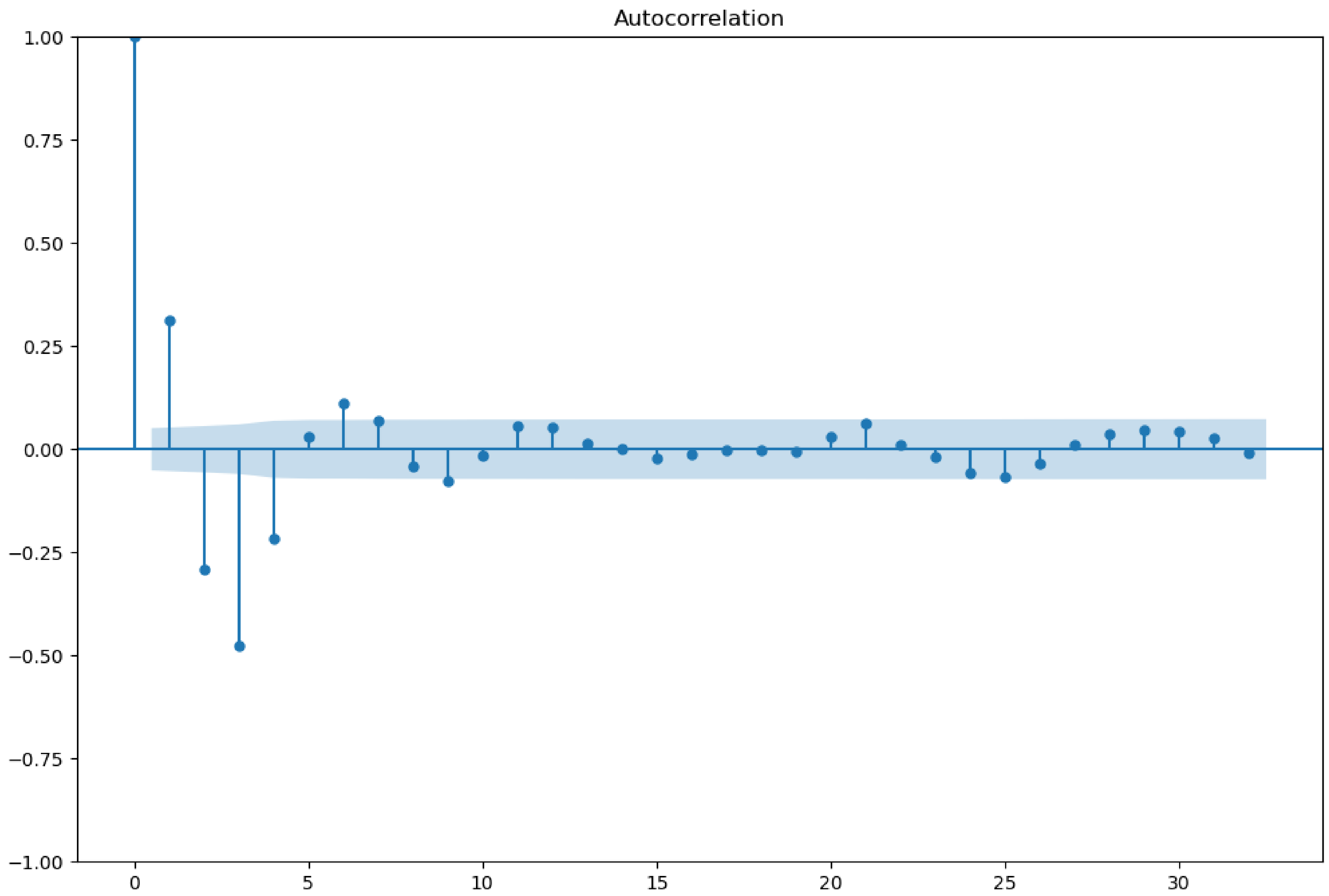
In [33]:
```
decomposeA = seasonal_decompose(df["temp_avg"], model = "additive", extrapolate_trend = "freq")
plt.rcParams["figure.figsize"] = (12, 8);
decomposeA.plot().suptitle("Additive Decomposition", fontsize=16)
```

Out[33]: Text(0.5, 0.98, 'Additive Decomposition')

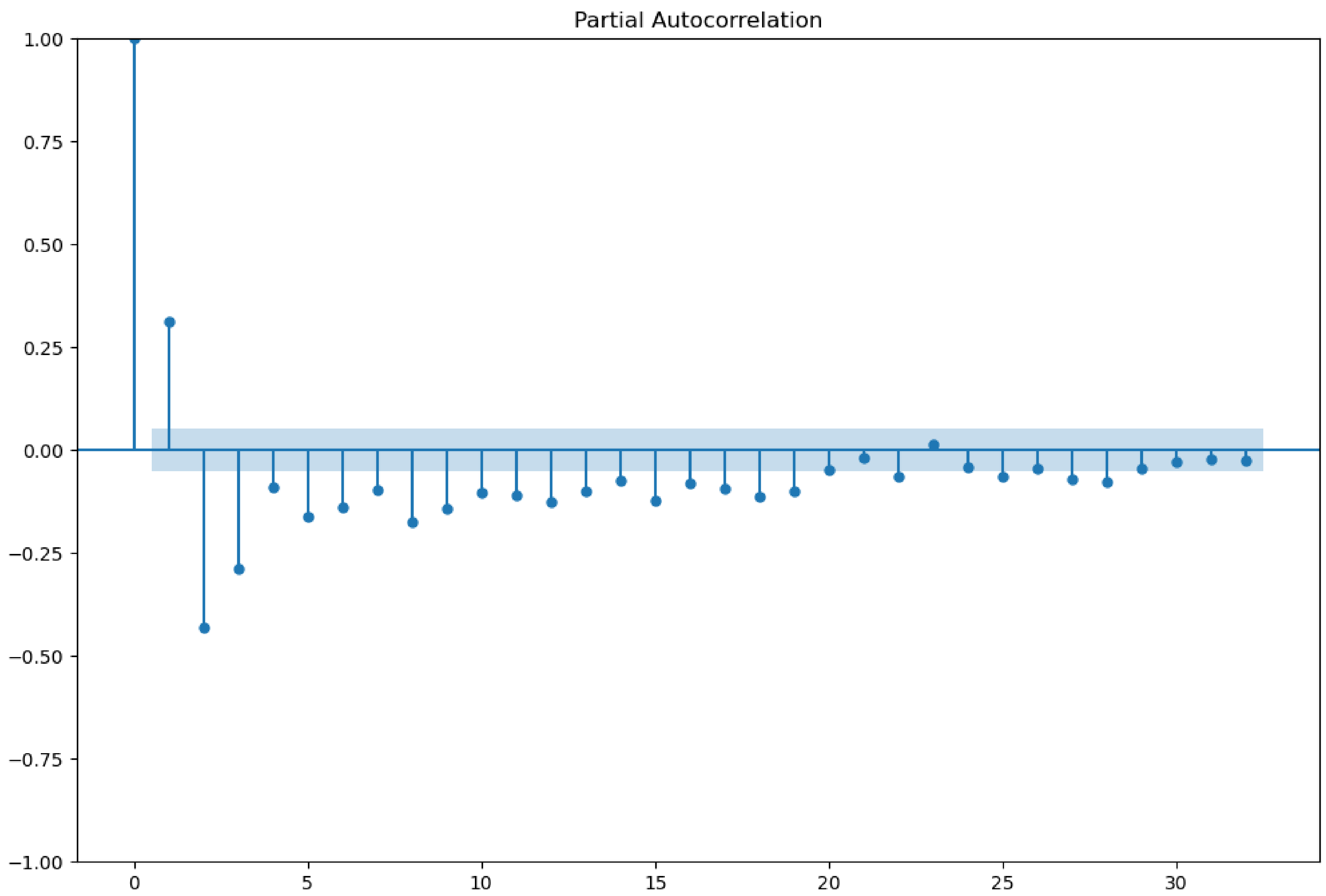


Residuals of additive decompostion is randomly scattering around 0 with constant variance, thus there is no obvious pattern in the residuals.

In [34]:
```
plot_acf(decomposeA.resid);
```



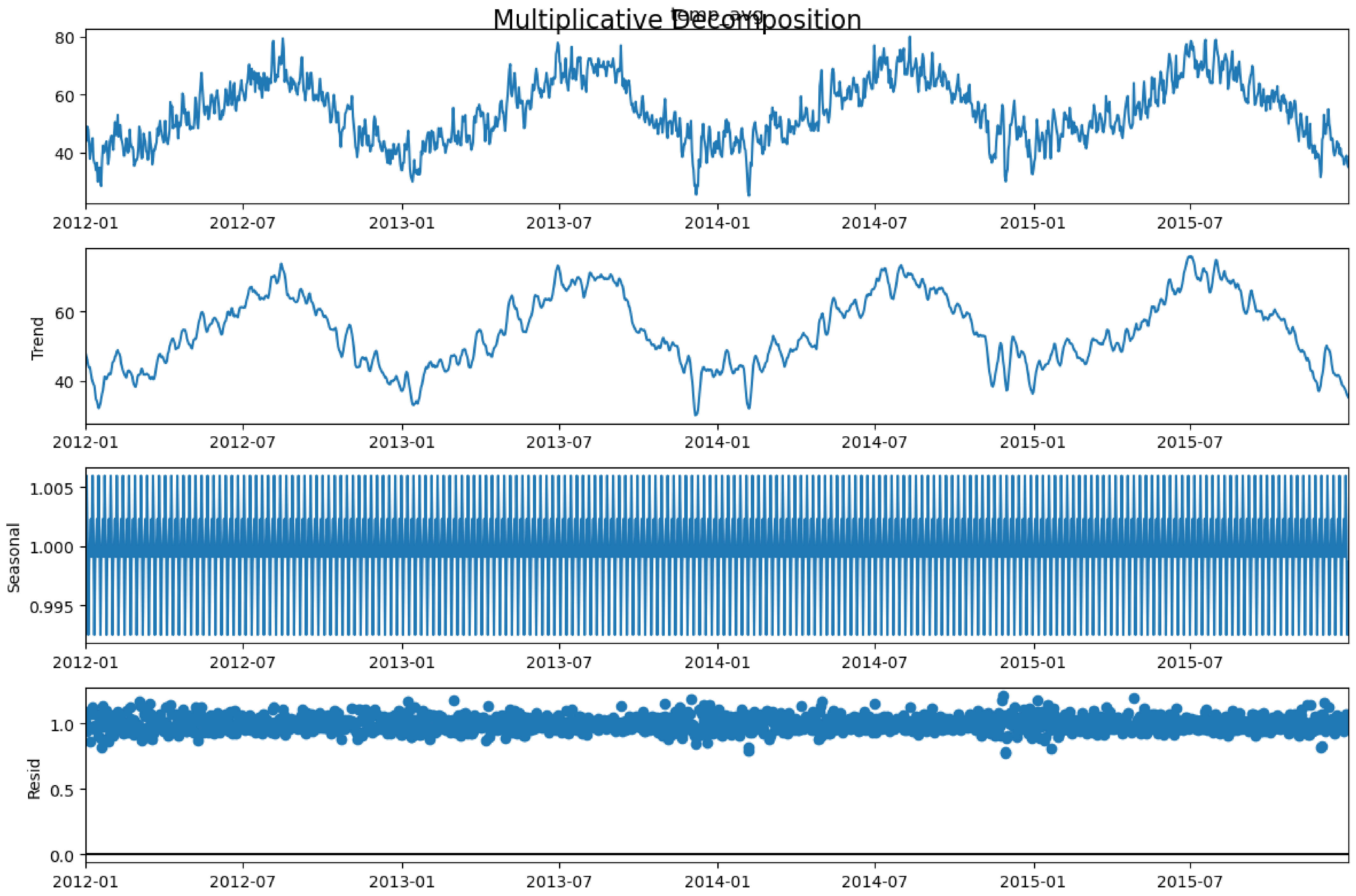
In [35]:
```
plot_pacf(decomposeA.resid, method = "ywm");
```



From acf plot, we could see that the autocorrelation between first four lags are statistically significant from the spikes. And from pacf plot, the autocorrelation gradually decreases as lag increases.

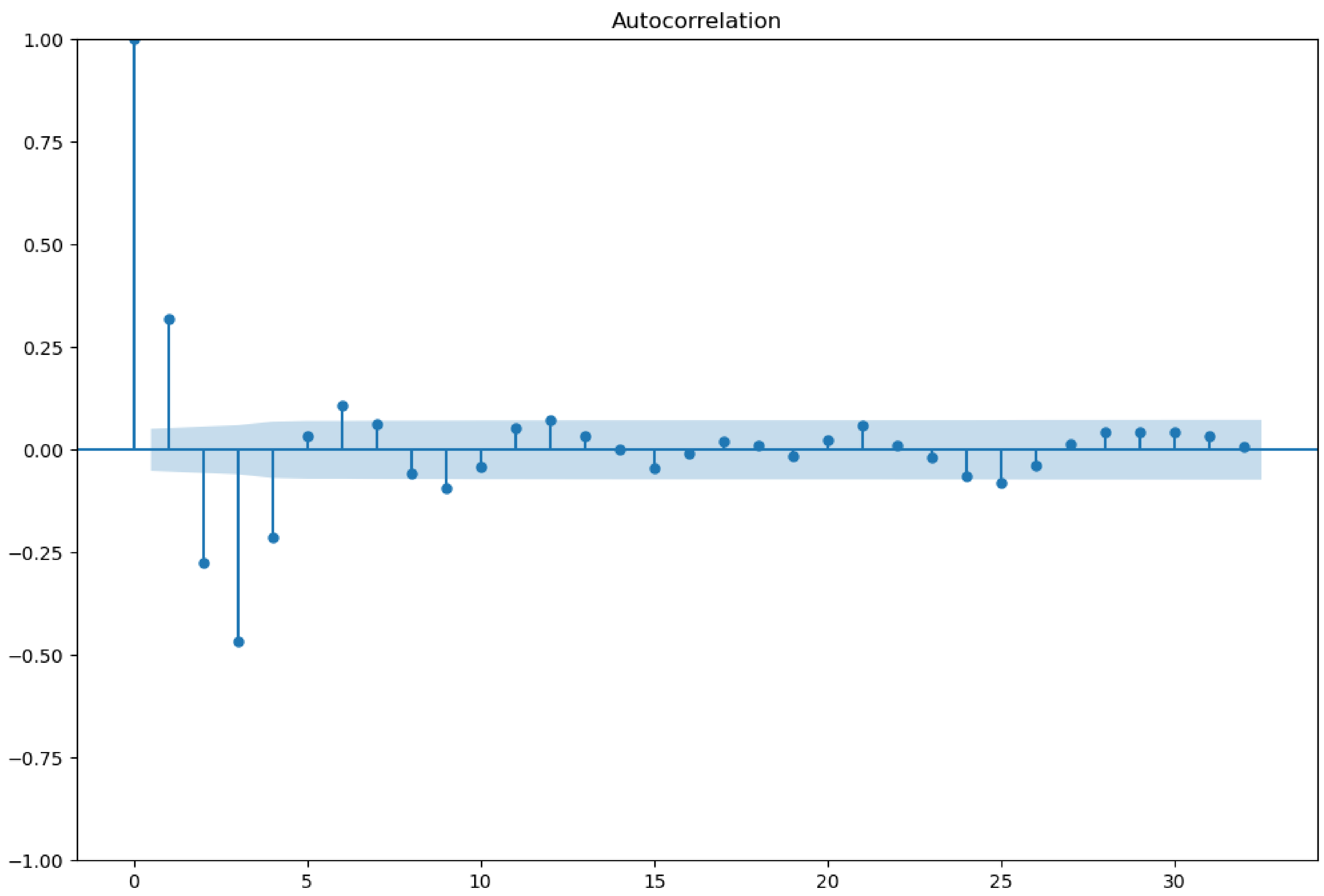
In [36]:
```
decomposeM = seasonal_decompose(df["temp_avg"], model = "multiplicative", extrapolate_trend = "freq")
plt.rcParams["figure.figsize"] = (12, 8);
decomposeM.plot().suptitle("Multiplicative Decomposition", fontsize=16)
```

Out[36]: Text(0.5, 0.98, 'Multiplicative Decomposition')

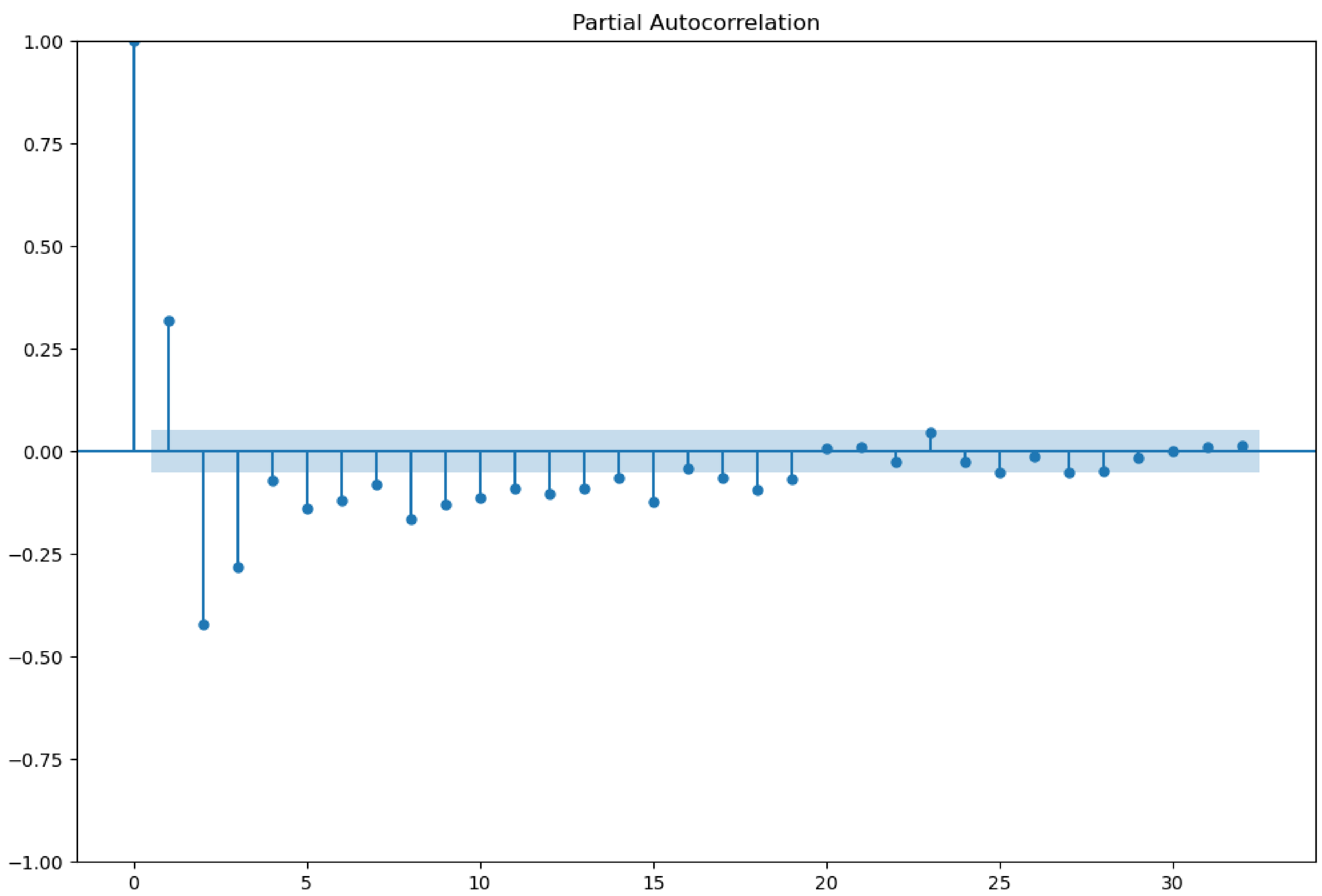


The residuals are scattering around 1 for multiplicative decomposition, with constant variance.

In [37]:
```
plot_acf(decomposeM.resid);
```



In [38]:
```
plot_pacf(decomposeM.resid, method = "ywm");
```



Similar to the additive decomposition, from acf plot, we could see that the autocorrelation between the first four lags are statistically significant from the spikes. And from pacf plot, autocorrelation gradually decreases as lag increases.

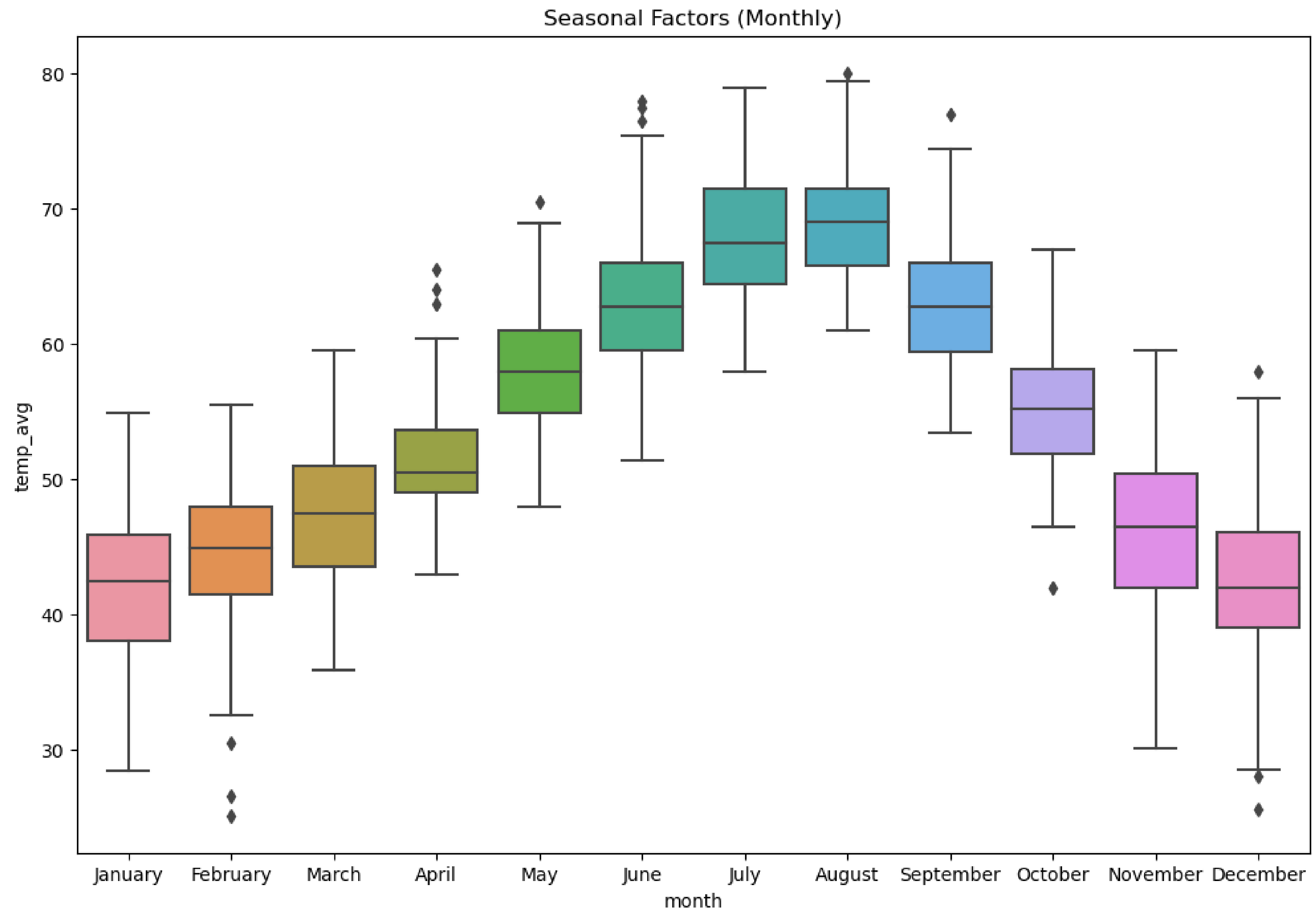
Additive decomposition is better. Firstly, residuals of additive decomposition are randomly scattering around 0 with constant variance, following White Noise distribution. Secondly, the scale or magnitude of the volatility of average temperature is constant over time, implying that additive decomposition would be more appropriate for the series data.

From the residual plots of two decompositions, we could see that they are similar, which means the models for cycles would be similar for both decompositions. Because they have similar acf plots with first four lags statistically significant, and also similar pacf plots with gradually decreasing pattern. Therefore, me might consider using Moving Average (MA) model to model the cycles for the time series data.

We include monthly seasonal factors and daily seasoanl factors plots below. We could see the average temprature is increasing from Jan. to Aug. every year, and then decreasing from Aug. to Dec from both plots. And during each year, the lowest temperature is around 40 Fahrenheit when in Jan and Dec, and that the highest temperature is around 70 Fahrenheit when in Aug.

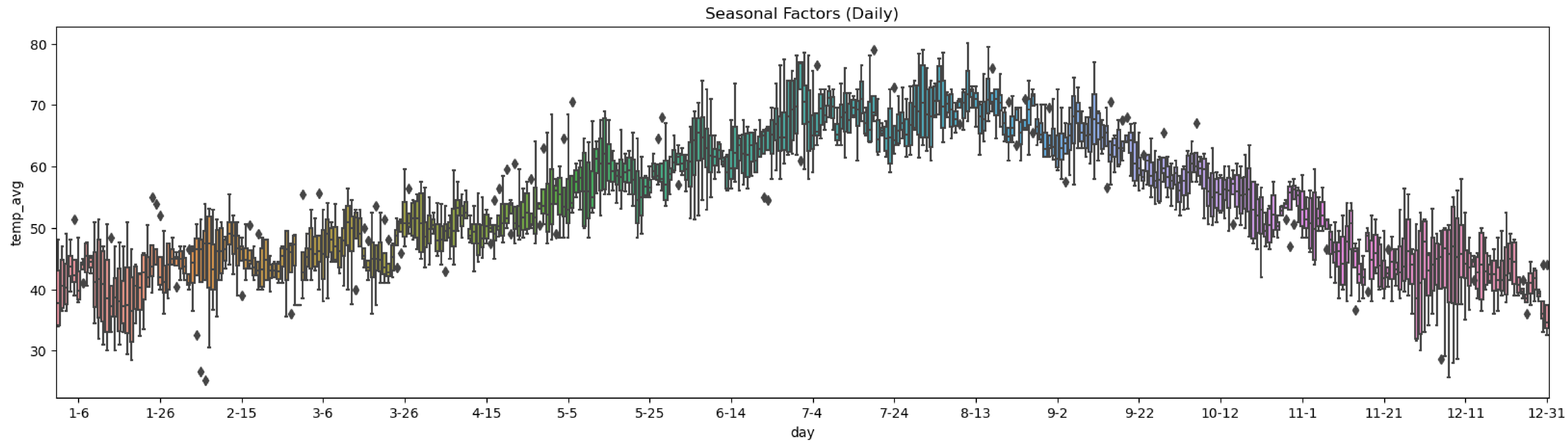
In [39]:
```
df["month"] = df.index.month_name()
sns.boxplot(x='month', y='temp_avg', data=df)
plt.title("Seasonal Factors (Monthly)")
```

Out[39]:
```
Text(0.5, 1.0, 'Seasonal Factors (Monthly)')
```



In [40]:
```
df["day"] = [str(i.month) + "-" + str(i.day) for i in df.index]
plt.figure(figsize = (20, 5))
sns.boxplot(x='day', y='temp_avg', data=df)
plt.xticks(range(5, 366, 20))
plt.title("Seasonal Factors (Daily)")
```

Out[40]:
```
Text(0.5, 1.0, 'Seasonal Factors (Daily)')
```



Preferred model: $Y_t = 138.37 + 0.0096t - 0.0000043t^2 - 193.2119 sin(\frac{2\pi t}{365}) - 96.4567 cos(\frac{2\pi t}{365}) + \sum_{i=1}^{365} \alpha_i D_i$

For the forecast plot, we include the mean value of the one-year ahead forecast with 95% confidence interval and 95% prediction interval following the original time series data in below.

In [41]:
```
model = smf.ols("temp_avg ~ t + I(t**2) + sint + cost + day", df).fit()
model.summary()
```

Out[41]:

OLS Regression Results

Dep. Variable:	temp_avg	**R-squared:**	0.848
Model:	OLS	**Adj. R-squared:**	0.797
Method:	Least Squares	**F-statistic:**	16.54
Date:	Sat, 22 Apr 2023	**Prob (F-statistic):**	1.89e-283
Time:	17:12:07	**Log-Likelihood:**	-4170.9
No. Observations:	1461	**AIC:**	9082.
Df Residuals:	1091	**BIC:**	1.104e+04
Df Model:	369		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	138.3722	132.978	1.041	0.298	-122.550	399.294
day[T.1-10]	30.6711	12.508	2.452	0.014	6.129	55.213
day[T.1-11]	33.3405	13.661	2.441	0.015	6.536	60.145
day[T.1-12]	34.5078	14.794	2.333	0.020	5.480	63.535
day[T.1-13]	36.7598	15.905	2.311	0.021	5.552	67.968
day[T.1-14]	39.1956	16.993	2.307	0.021	5.852	72.539
day[T.1-15]	41.6119	18.059	2.304	0.021	6.179	77.045
day[T.1-16]	44.3453	19.100	2.322	0.020	6.869	81.822
day[T.1-17]	47.0125	20.118	2.337	0.020	7.539	86.486
day[T.1-18]	49.9053	21.111	2.364	0.018	8.482	91.329
day[T.1-19]	50.6153	22.081	2.292	0.022	7.289	93.942
day[T.1-2]	5.0092	3.733	1.342	0.180	-2.315	12.333
day[T.1-20]	56.0493	23.028	2.434	0.015	10.865	101.233
day[T.1-21]	57.5664	23.951	2.403	0.016	10.571	104.562
day[T.1-22]	62.3886	24.852	2.510	0.012	13.626	111.151
day[T.1-23]	67.2975	25.730	2.616	0.009	16.812	117.783
day[T.1-24]	71.4149	26.586	2.686	0.007	19.249	123.581
day[T.1-25]	74.7401	27.422	2.726	0.007	20.935	128.545
day[T.1-26]	74.5050	28.236	2.639	0.008	19.101	129.909
day[T.1-27]	74.7364	29.031	2.574	0.010	17.773	131.700
day[T.1-28]	78.3810	29.807	2.630	0.009	19.895	136.867
day[T.1-29]	82.5808	30.564	2.702	0.007	22.609	142.553
day[T.1-3]	8.5921	4.480	1.918	0.055	-0.199	17.383
day[T.1-30]	83.1502	31.304	2.656	0.008	21.726	144.574
day[T.1-31]	85.6459	32.028	2.674	0.008	22.803	148.489
day[T.1-4]	14.0329	5.478	2.562	0.011	3.284	24.782
day[T.1-5]	17.1680	6.596	2.603	0.009	4.226	30.110
day[T.1-6]	18.0414	7.767	2.323	0.020	2.801	33.282
day[T.1-7]	23.1324	8.958	2.582	0.010	5.555	40.710
day[T.1-8]	27.4675	10.151	2.706	0.007	7.549	47.386
day[T.1-9]	29.9432	11.337	2.641	0.008	7.699	52.187
day[T.10-1]	-275.7736	204.123	-1.351	0.177	-676.291	124.744
day[T.10-10]	-260.1226	183.753	-1.416	0.157	-620.671	100.426
day[T.10-11]	-258.3086	181.406	-1.424	0.155	-614.253	97.636
day[T.10-12]	-255.9038	179.045	-1.429	0.153	-607.216	95.408
day[T.10-13]	-253.0665	176.671	-1.432	0.152	-599.720	93.587
day[T.10-14]	-251.2824	174.284	-1.442	0.150	-593.251	90.686
day[T.10-15]	-248.1222	171.884	-1.444	0.149	-585.383	89.138
day[T.10-16]	-245.7467	169.473	-1.450	0.147	-578.276	86.782
day[T.10-17]	-243.9764	167.050	-1.460	0.144	-571.752	83.799
day[T.10-18]	-240.8773	164.617	-1.463	0.144	-563.880	82.125
day[T.10-19]	-237.8901	162.175	-1.467	0.143	-556.101	80.320
day[T.10-2]	-276.3563	201.934	-1.369	0.171	-672.578	119.866
day[T.10-20]	-238.5705	159.724	-1.494	0.136	-551.971	74.830
day[T.10-21]	-236.7544	157.264	-1.505	0.132	-545.329	71.820
day[T.10-22]	-234.7375	154.797	-1.516	0.130	-538.471	68.996
day[T.10-23]	-233.4431	152.323	-1.533	0.126	-532.322	65.436
day[T.10-24]	-229.9697	149.843	-1.535	0.125	-523.982	64.043
day[T.10-25]	-224.7005	147.357	-1.525	0.128	-513.835	64.434
day[T.10-26]	-224.1615	144.866	-1.547	0.122	-508.408	60.085
day[T.10-27]	-218.7008	142.371	-1.536	0.125	-498.052	60.650
day[T.10-28]	-214.9796	139.872	-1.537	0.125	-489.428	59.469
day[T.10-29]	-211.4010	137.370	-1.539	0.124	-480.941	58.139
day[T.10-3]	-273.0579	199.724	-1.367	0.172	-664.945	118.829
day[T.10-30]	-207.0661	134.867	-1.535	0.125	-471.694	57.561
day[T.10-31]	-204.6983	132.361	-1.547	0.122	-464.410	55.014
day[T.10-4]	-269.9064	197.496	-1.367	0.172	-657.421	117.608
day[T.10-5]	-265.7549	195.248	-1.361	0.174	-648.860	117.350
day[T.10-6]	-263.2589	192.983	-1.364	0.173	-641.919	115.401
day[T.10-7]	-262.8466	190.700	-1.378	0.168	-637.027	111.334
day[T.10-8]	-262.1336	188.400	-1.391	0.164	-631.801	107.534
day[T.10-9]	-262.4028	186.084	-1.410	0.159	-627.526	102.721
day[T.11-1]	-202.8134	129.855	-1.562	0.119	-457.608	51.981
day[T.11-10]	-181.2575	107.376	-1.688	0.092	-391.945	29.430
day[T.11-11]	-177.8202	104.899	-1.695	0.090	-383.647	28.007
day[T.11-12]	-172.5815	102.429	-1.685	0.092	-373.561	28.398
day[T.11-13]	-167.3652	99.965	-1.674	0.094	-363.511	28.781
day[T.11-14]	-168.3596	97.510	-1.727	0.085	-359.688	22.969
day[T.11-15]	-166.2058	95.063	-1.748	0.081	-352.733	20.322
day[T.11-16]	-163.4025	92.625	-1.764	0.078	-345.147	18.342
day[T.11-17]	-155.0680	90.198	-1.719	0.086	-332.048	21.912
day[T.11-18]	-153.3985	87.780	-1.748	0.081	-325.635	18.838
day[T.11-19]	-148.2026	85.374	-1.736	0.083	-315.718	19.312
day[T.11-2]	-201.2999	127.349	-1.581	0.114	-451.177	48.577
day[T.11-20]	-147.0412	82.979	-1.772	0.077	-309.857	15.775

day[T.11-21]	-144.5381	80.596	-1.793	0.073	-302.680	13.603
day[T.11-22]	-140.7842	78.227	-1.800	0.072	-294.276	12.708
day[T.11-23]	-136.2532	75.871	-1.796	0.073	-285.123	12.616
day[T.11-24]	-133.0610	73.529	-1.810	0.071	-277.335	11.213
day[T.11-25]	-128.5764	71.202	-1.806	0.071	-268.285	11.132
day[T.11-26]	-123.2279	68.890	-1.789	0.074	-258.400	11.944
day[T.11-27]	-118.6590	66.594	-1.782	0.075	-249.326	12.008
day[T.11-28]	-119.1010	64.315	-1.852	0.064	-245.296	7.094
day[T.11-29]	-118.2098	62.053	-1.905	0.057	-239.966	3.546
day[T.11-3]	-198.8087	124.844	-1.592	0.112	-443.770	46.152
day[T.11-30]	-113.1291	59.808	-1.892	0.059	-230.480	4.222
day[T.11-4]	-195.6784	122.339	-1.599	0.110	-435.725	44.369
day[T.11-5]	-191.2798	119.837	-1.596	0.111	-426.416	43.857
day[T.11-6]	-187.3240	117.337	-1.596	0.111	-417.555	42.908
day[T.11-7]	-186.4443	114.840	-1.624	0.105	-411.777	38.888
day[T.11-8]	-186.0544	112.347	-1.656	0.098	-406.496	34.387
day[T.11-9]	-183.7926	109.859	-1.673	0.095	-399.352	31.767
day[T.12-1]	-105.5924	57.581	-1.834	0.067	-218.575	7.390
day[T.12-10]	-70.4378	38.450	-1.832	0.067	-145.883	5.007
day[T.12-11]	-69.5425	36.435	-1.909	0.057	-141.034	1.949
day[T.12-12]	-66.8333	34.445	-1.940	0.053	-134.418	0.752
day[T.12-13]	-62.1989	32.478	-1.915	0.056	-125.926	1.528
day[T.12-14]	-60.0278	30.537	-1.966	0.050	-119.946	-0.110
day[T.12-15]	-55.2587	28.622	-1.931	0.054	-111.419	0.901
day[T.12-16]	-51.6500	26.733	-1.932	0.054	-104.104	0.804
day[T.12-17]	-48.7755	24.871	-1.961	0.050	-97.576	0.025
day[T.12-18]	-46.1635	23.037	-2.004	0.045	-91.366	-0.961
day[T.12-19]	-42.4654	21.232	-2.000	0.046	-84.126	-0.805
day[T.12-2]	-105.3435	55.373	-1.902	0.057	-213.994	3.307
day[T.12-20]	-37.9970	19.457	-1.953	0.051	-76.174	0.180
day[T.12-21]	-33.5694	17.713	-1.895	0.058	-68.324	1.185
day[T.12-22]	-28.6662	16.001	-1.791	0.073	-60.063	2.730
day[T.12-23]	-26.3035	14.325	-1.836	0.067	-54.411	1.804
day[T.12-24]	-26.1673	12.687	-2.063	0.039	-51.060	-1.274
day[T.12-25]	-23.5561	11.091	-2.124	0.034	-45.319	-1.793
day[T.12-26]	-21.3734	9.547	-2.239	0.025	-40.105	-2.642
day[T.12-27]	-15.5028	8.064	-1.922	0.055	-31.326	0.320
day[T.12-28]	-11.8403	6.666	-1.776	0.076	-24.920	1.240
day[T.12-29]	-10.5894	5.394	-1.963	0.050	-21.173	-0.006
day[T.12-3]	-98.9058	53.185	-1.860	0.063	-203.262	5.451
day[T.12-30]	-9.0960	4.329	-2.101	0.036	-17.590	-0.602
day[T.12-31]	-6.6637	3.621	-1.840	0.066	-13.769	0.442
day[T.12-4]	-96.1805	51.016	-1.885	0.060	-196.282	3.921
day[T.12-5]	-93.5912	48.868	-1.915	0.056	-189.477	2.295
day[T.12-6]	-88.8665	46.741	-1.901	0.058	-180.578	2.845
day[T.12-7]	-86.4399	44.635	-1.937	0.053	-174.019	1.139
day[T.12-8]	-80.3501	42.551	-1.888	0.059	-163.840	3.140
day[T.12-9]	-77.1232	40.489	-1.905	0.057	-156.568	2.322
day[T.2-1]	86.9850	32.736	2.657	0.008	22.753	151.217
day[T.2-10]	104.2496	38.585	2.702	0.007	28.540	179.959
day[T.2-11]	105.6255	39.200	2.695	0.007	28.710	182.541
day[T.2-12]	110.0243	39.812	2.764	0.006	31.908	188.141
day[T.2-13]	109.9306	40.424	2.719	0.007	30.614	189.247
day[T.2-14]	109.0115	41.036	2.656	0.008	28.493	189.530
day[T.2-15]	109.3817	41.651	2.626	0.009	27.657	191.106
day[T.2-16]	110.7033	42.269	2.619	0.009	27.766	193.641
day[T.2-17]	112.1661	42.893	2.615	0.009	28.005	196.327
day[T.2-18]	111.1821	43.523	2.555	0.011	25.784	196.580
day[T.2-19]	111.9587	44.161	2.535	0.011	25.308	198.610
day[T.2-2]	87.0094	33.429	2.603	0.009	21.417	152.602
day[T.2-20]	112.7855	44.810	2.517	0.012	24.862	200.709
day[T.2-21]	114.5624	45.469	2.520	0.012	25.345	203.780
day[T.2-22]	113.2615	46.142	2.455	0.014	22.725	203.798
day[T.2-23]	114.2603	46.828	2.440	0.015	22.376	206.144
day[T.2-24]	114.5659	47.531	2.410	0.016	21.304	207.828
day[T.2-25]	116.9682	48.250	2.424	0.016	22.295	211.641
day[T.2-26]	117.1922	48.988	2.392	0.017	21.072	213.313
day[T.2-27]	116.9250	49.745	2.350	0.019	19.318	214.532
day[T.2-28]	118.5967	50.523	2.347	0.019	19.463	217.731
day[T.2-29]	113.9026	51.072	2.230	0.026	13.691	214.114
day[T.2-3]	89.7235	34.109	2.631	0.009	22.798	156.649
day[T.2-4]	92.5617	34.776	2.662	0.008	24.327	160.797
day[T.2-5]	93.0035	35.432	2.625	0.009	23.481	162.526
day[T.2-6]	95.4360	36.078	2.645	0.008	24.646	166.225
day[T.2-7]	98.1260	36.715	2.673	0.008	26.087	170.165
day[T.2-8]	99.0707	37.344	2.653	0.008	25.797	172.345
day[T.2-9]	101.3971	37.967	2.671	0.008	26.900	175.894
day[T.3-1]	118.5314	51.528	2.300	0.022	17.426	219.637
day[T.3-10]	118.8443	59.963	1.982	0.048	1.188	236.500
day[T.3-11]	119.9548	61.046	1.965	0.050	0.174	239.735
day[T.3-12]	121.7443	62.160	1.959	0.050	-0.221	243.710
day[T.3-13]	119.6228	63.305	1.890	0.059	-4.590	243.836
day[T.3-14]	120.1156	64.482	1.863	0.063	-6.407	246.638
day[T.3-15]	119.8702	65.691	1.825	0.068	-9.024	248.765
day[T.3-16]	115.7820	66.932	1.730	0.084	-15.548	247.112
day[T.3-17]	113.0035	68.205	1.657	0.098	-20.824	246.831

day[T.3-18]	112.6601	69.510	1.621	0.105	-23.729	249.049
day[T.3-19]	112.3894	70.848	1.586	0.113	-26.624	251.403
day[T.3-2]	119.1144	52.358	2.275	0.023	16.380	221.849
day[T.3-20]	111.9218	72.217	1.550	0.121	-29.779	253.622
day[T.3-21]	110.0200	73.618	1.494	0.135	-34.430	254.470
day[T.3-22]	107.6968	75.051	1.435	0.152	-39.564	254.958
day[T.3-23]	107.9226	76.515	1.410	0.159	-42.211	258.056
day[T.3-24]	109.8427	78.010	1.408	0.159	-43.224	262.909
day[T.3-25]	110.1724	79.536	1.385	0.166	-45.888	266.232
day[T.3-26]	111.1396	81.091	1.371	0.171	-47.973	270.252
day[T.3-27]	109.4598	82.677	1.324	0.186	-52.764	271.683
day[T.3-28]	107.6532	84.291	1.277	0.202	-57.738	273.045
day[T.3-29]	106.0354	85.935	1.234	0.218	-62.581	274.652
day[T.3-3]	120.8485	53.213	2.271	0.023	16.437	225.260
day[T.3-30]	104.2243	87.606	1.190	0.234	-67.672	276.120
day[T.3-31]	100.8704	89.306	1.129	0.259	-74.360	276.101
day[T.3-4]	120.5612	54.093	2.229	0.026	14.423	226.700
day[T.3-5]	121.0199	55.000	2.200	0.028	13.102	228.938
day[T.3-6]	119.5471	55.935	2.137	0.033	9.795	229.299
day[T.3-7]	121.4303	56.898	2.134	0.033	9.789	233.072
day[T.3-8]	121.4947	57.890	2.099	0.036	7.907	235.082
day[T.3-9]	121.4051	58.911	2.061	0.040	5.813	236.997
day[T.4-1]	99.5291	91.032	1.093	0.274	-79.088	278.146
day[T.4-10]	85.4581	107.676	0.794	0.428	-125.817	296.734
day[T.4-11]	79.3431	109.636	0.724	0.469	-135.777	294.464
day[T.4-12]	76.5747	111.614	0.686	0.493	-142.429	295.578
day[T.4-13]	74.2734	113.612	0.654	0.513	-148.649	297.196
day[T.4-14]	72.2599	115.627	0.625	0.532	-154.617	299.137
day[T.4-15]	70.9175	117.659	0.603	0.547	-159.947	301.782
day[T.4-16]	67.9744	119.708	0.568	0.570	-166.909	302.858
day[T.4-17]	65.0738	121.772	0.534	0.593	-173.860	304.007
day[T.4-18]	64.5114	123.850	0.521	0.603	-178.500	307.523
day[T.4-19]	62.3955	125.943	0.495	0.620	-184.722	309.513
day[T.4-2]	98.8059	92.785	1.065	0.287	-83.250	280.862
day[T.4-20]	61.0219	128.048	0.477	0.634	-190.227	312.271
day[T.4-21]	57.9164	130.166	0.445	0.656	-197.487	313.320
day[T.4-22]	54.7898	132.295	0.414	0.679	-204.792	314.371
day[T.4-23]	51.6653	134.435	0.384	0.701	-212.115	315.445
day[T.4-24]	48.9263	136.584	0.358	0.720	-219.072	316.924
day[T.4-25]	46.8436	138.743	0.338	0.736	-225.390	319.077
day[T.4-26]	43.3032	140.910	0.307	0.759	-233.182	319.789
day[T.4-27]	41.8382	143.084	0.292	0.770	-238.914	322.590
day[T.4-28]	38.1972	145.265	0.263	0.793	-246.834	323.228
day[T.4-29]	37.1960	147.452	0.252	0.801	-252.126	326.518
day[T.4-3]	95.2588	94.563	1.007	0.314	-90.287	280.805
day[T.4-30]	33.3681	149.644	0.223	0.824	-260.255	326.991
day[T.4-4]	93.0510	96.366	0.966	0.334	-96.034	282.136
day[T.4-5]	91.5303	98.194	0.932	0.351	-101.141	284.202
day[T.4-6]	89.9099	100.046	0.899	0.369	-106.395	286.215
day[T.4-7]	89.2704	101.921	0.876	0.381	-110.713	289.254
day[T.4-8]	89.9499	103.818	0.866	0.386	-113.755	293.655
day[T.4-9]	86.4589	105.737	0.818	0.414	-121.011	293.929
day[T.5-1]	32.0017	151.840	0.211	0.833	-265.930	329.933
day[T.5-10]	2.6511	171.664	0.015	0.988	-334.177	339.479
day[T.5-11]	0.6179	173.858	0.004	0.997	-340.517	341.753
day[T.5-12]	0.2243	176.049	0.001	0.999	-345.208	345.657
day[T.5-13]	-3.6135	178.234	-0.020	0.984	-353.334	346.107
day[T.5-14]	-5.0895	180.413	-0.028	0.977	-359.086	348.907
day[T.5-15]	-9.7378	182.586	-0.053	0.957	-367.997	348.521
day[T.5-16]	-14.9922	184.751	-0.081	0.935	-377.499	347.515
day[T.5-17]	-19.8843	186.908	-0.106	0.915	-386.624	346.855
day[T.5-18]	-22.4329	189.056	-0.119	0.906	-393.387	348.521
day[T.5-19]	-25.9221	191.194	-0.136	0.892	-401.072	349.227
day[T.5-2]	27.5403	154.040	0.179	0.858	-274.707	329.788
day[T.5-20]	-29.2933	193.322	-0.152	0.880	-408.618	350.031
day[T.5-21]	-33.4678	195.438	-0.171	0.864	-416.946	350.010
day[T.5-22]	-40.2447	197.543	-0.204	0.839	-427.852	347.363
day[T.5-23]	-44.2679	199.636	-0.222	0.825	-435.981	347.445
day[T.5-24]	-46.3012	201.714	-0.230	0.818	-442.093	349.491
day[T.5-25]	-49.6962	203.779	-0.244	0.807	-449.540	350.148
day[T.5-26]	-50.7617	205.830	-0.247	0.805	-454.628	353.105
day[T.5-27]	-54.4466	207.864	-0.262	0.793	-462.306	353.413
day[T.5-28]	-57.7573	209.883	-0.275	0.783	-469.578	354.063
day[T.5-29]	-62.8528	211.885	-0.297	0.767	-478.601	352.896
day[T.5-3]	24.4849	156.242	0.157	0.876	-282.083	331.053
day[T.5-30]	-66.5370	213.870	-0.311	0.756	-486.179	353.105
day[T.5-31]	-67.5262	215.836	-0.313	0.754	-491.027	355.974
day[T.5-4]	21.3739	158.446	0.135	0.893	-289.519	332.266
day[T.5-5]	18.9057	160.651	0.118	0.906	-296.313	334.125
day[T.5-6]	17.6663	162.856	0.108	0.914	-301.880	337.212
day[T.5-7]	14.2367	165.061	0.086	0.931	-309.635	338.109
day[T.5-8]	10.8678	167.264	0.065	0.948	-317.328	339.063
day[T.5-9]	6.5481	169.465	0.039	0.969	-325.966	339.063
day[T.6-1]	-71.7144	217.784	-0.329	0.742	-499.036	355.608
day[T.6-10]	-104.5218	234.362	-0.446	0.656	-564.373	355.329
day[T.6-11]	-107.6315	236.087	-0.456	0.649	-570.867	355.604

day[T.6-12]	-112.3980	237.787	-0.473	0.637	-578.969	354.173
day[T.6-13]	-118.1004	239.460	-0.493	0.622	-587.955	351.754
day[T.6-14]	-119.8774	241.107	-0.497	0.619	-592.963	353.208
day[T.6-15]	-120.5630	242.727	-0.497	0.619	-596.827	355.701
day[T.6-16]	-126.0511	244.318	-0.516	0.606	-605.438	353.336
day[T.6-17]	-129.5457	245.882	-0.527	0.598	-612.001	352.909
day[T.6-18]	-133.5882	247.417	-0.540	0.589	-619.054	351.878
day[T.6-19]	-136.1524	248.922	-0.547	0.585	-624.573	352.268
day[T.6-2]	-75.6804	219.712	-0.344	0.731	-506.786	355.425
day[T.6-20]	-140.7473	250.398	-0.562	0.574	-632.063	350.568
day[T.6-21]	-142.3995	251.843	-0.565	0.572	-636.551	351.752
day[T.6-22]	-146.8227	253.257	-0.580	0.562	-643.749	350.104
day[T.6-23]	-150.0110	254.641	-0.589	0.556	-649.652	349.630
day[T.6-24]	-152.1658	255.992	-0.594	0.552	-654.458	350.127
day[T.6-25]	-153.3761	257.311	-0.596	0.551	-658.257	351.505
day[T.6-26]	-157.5784	258.598	-0.609	0.542	-664.984	349.828
day[T.6-27]	-160.1365	259.852	-0.616	0.538	-670.002	349.729
day[T.6-28]	-162.3096	261.072	-0.622	0.534	-674.569	349.950
day[T.6-29]	-164.7941	262.258	-0.628	0.530	-679.381	349.793
day[T.6-3]	-80.8182	221.620	-0.365	0.715	-515.668	354.031
day[T.6-30]	-167.2740	263.410	-0.635	0.526	-684.122	349.574
day[T.6-4]	-83.8866	223.507	-0.375	0.707	-522.439	354.666
day[T.6-5]	-85.2446	225.373	-0.378	0.705	-527.459	356.970
day[T.6-6]	-87.8385	227.217	-0.387	0.699	-533.671	357.994
day[T.6-7]	-90.4298	229.039	-0.395	0.693	-539.836	358.977
day[T.6-8]	-95.7623	230.837	-0.415	0.678	-548.697	357.173
day[T.6-9]	-99.9875	232.612	-0.430	0.667	-556.405	356.429
day[T.7-1]	-167.6332	264.527	-0.634	0.526	-686.673	351.407
day[T.7-10]	-203.1477	272.963	-0.744	0.457	-738.741	332.445
day[T.7-11]	-203.7809	273.715	-0.745	0.457	-740.848	333.287
day[T.7-12]	-206.2431	274.428	-0.752	0.452	-744.710	332.224
day[T.7-13]	-209.7910	275.103	-0.763	0.446	-749.582	330.000
day[T.7-14]	-211.1387	275.739	-0.766	0.444	-752.178	329.900
day[T.7-15]	-215.6405	276.336	-0.780	0.435	-757.851	326.570
day[T.7-16]	-214.9479	276.894	-0.776	0.438	-758.254	328.358
day[T.7-17]	-222.0201	277.413	-0.800	0.424	-766.343	322.303
day[T.7-18]	-223.3788	277.892	-0.804	0.422	-768.642	321.884
day[T.7-19]	-223.5458	278.331	-0.803	0.422	-769.671	322.580
day[T.7-2]	-174.0384	265.610	-0.655	0.512	-695.202	347.125
day[T.7-20]	-231.6776	278.731	-0.831	0.406	-778.587	315.231
day[T.7-21]	-233.8461	279.090	-0.838	0.402	-781.460	313.768
day[T.7-22]	-237.7680	279.409	-0.851	0.395	-786.008	310.472
day[T.7-23]	-239.5050	279.688	-0.856	0.392	-788.292	309.282
day[T.7-24]	-241.3065	279.926	-0.862	0.389	-790.561	307.948
day[T.7-25]	-242.9241	280.124	-0.867	0.386	-792.567	306.719
day[T.7-26]	-244.0197	280.281	-0.871	0.384	-793.971	305.931
day[T.7-27]	-248.4175	280.397	-0.886	0.376	-798.597	301.762
day[T.7-28]	-249.2770	280.473	-0.889	0.374	-799.604	301.050
day[T.7-29]	-248.7125	280.507	-0.887	0.375	-799.108	301.683
day[T.7-3]	-179.4236	266.656	-0.673	0.501	-702.641	343.794
day[T.7-30]	-251.2682	280.501	-0.896	0.371	-801.651	299.114
day[T.7-31]	-252.0161	280.454	-0.899	0.369	-802.306	298.274
day[T.7-4]	-184.1702	267.667	-0.688	0.492	-709.371	341.031
day[T.7-5]	-185.6448	268.642	-0.691	0.490	-712.759	341.469
day[T.7-6]	-187.6940	269.581	-0.696	0.486	-716.649	341.261
day[T.7-7]	-191.0593	270.482	-0.706	0.480	-721.784	339.665
day[T.7-8]	-192.0723	271.347	-0.708	0.479	-724.493	340.348
day[T.7-9]	-196.4921	272.174	-0.722	0.470	-730.536	337.552
day[T.8-1]	-255.9281	280.365	-0.913	0.362	-806.045	294.188
day[T.8-10]	-270.7278	277.729	-0.975	0.330	-815.671	274.215
day[T.8-11]	-269.2152	277.232	-0.971	0.332	-813.183	274.753
day[T.8-12]	-271.6035	276.695	-0.982	0.327	-814.518	271.311
day[T.8-13]	-274.3147	276.118	-0.993	0.321	-816.096	267.467
day[T.8-14]	-278.7435	275.500	-1.012	0.312	-819.313	261.826
day[T.8-15]	-278.5671	274.842	-1.014	0.311	-817.847	260.712
day[T.8-16]	-276.2374	274.145	-1.008	0.314	-814.149	261.674
day[T.8-17]	-277.9418	273.408	-1.017	0.310	-814.407	258.523
day[T.8-18]	-280.7323	272.631	-1.030	0.303	-815.673	254.209
day[T.8-19]	-280.9412	271.815	-1.034	0.302	-814.281	252.399
day[T.8-2]	-258.9085	280.236	-0.924	0.356	-808.771	290.954
day[T.8-20]	-285.4757	270.960	-1.054	0.292	-817.138	246.187
day[T.8-21]	-287.8780	270.067	-1.066	0.287	-817.787	242.031
day[T.8-22]	-287.0229	269.134	-1.066	0.286	-815.102	241.056
day[T.8-23]	-286.8476	268.163	-1.070	0.285	-813.022	239.327
day[T.8-24]	-290.0294	267.154	-1.086	0.278	-814.224	234.165
day[T.8-25]	-288.6256	266.107	-1.085	0.278	-810.766	233.515
day[T.8-26]	-287.9236	265.023	-1.086	0.278	-807.936	232.089
day[T.8-27]	-286.7756	263.901	-1.087	0.277	-804.587	231.036
day[T.8-28]	-290.7841	262.742	-1.107	0.269	-806.322	224.754
day[T.8-29]	-291.6464	261.547	-1.115	0.265	-804.838	221.546
day[T.8-3]	-257.9419	280.066	-0.921	0.357	-807.471	291.587
day[T.8-30]	-293.7724	260.315	-1.129	0.259	-804.547	217.002
day[T.8-31]	-293.7421	259.047	-1.134	0.257	-802.029	214.545
day[T.8-4]	-257.1675	279.855	-0.919	0.358	-806.282	291.947
day[T.8-5]	-261.8499	279.602	-0.937	0.349	-810.469	286.769
day[T.8-6]	-263.9560	279.309	-0.945	0.345	-812.000	284.088

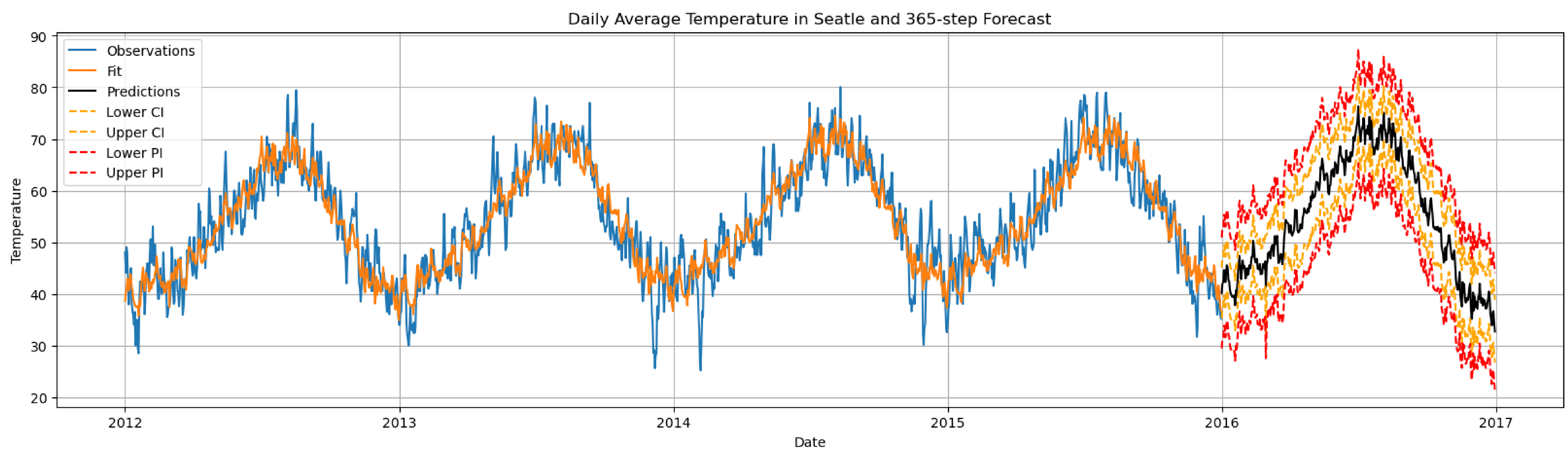
day[T.8-7]	-266.9054	278.975	-0.957	0.339	-814.294	280.483
day[T.8-8]	-269.5499	278.600	-0.968	0.334	-816.203	277.103
day[T.8-9]	-269.2792	278.185	-0.968	0.333	-815.117	276.559
day[T.9-1]	-293.7604	257.743	-1.140	0.255	-799.489	211.969
day[T.9-10]	-291.7896	244.465	-1.194	0.233	-771.465	187.886
day[T.9-11]	-288.8068	242.826	-1.189	0.235	-765.265	187.651
day[T.9-12]	-289.8782	241.155	-1.202	0.230	-763.057	183.301
day[T.9-13]	-290.4813	239.453	-1.213	0.225	-760.321	179.359
day[T.9-14]	-290.3014	237.721	-1.221	0.222	-756.743	176.140
day[T.9-15]	-290.9812	235.958	-1.233	0.218	-753.965	172.003
day[T.9-16]	-290.3159	234.167	-1.240	0.215	-749.785	169.153
day[T.9-17]	-288.8007	232.347	-1.243	0.214	-744.698	167.096
day[T.9-18]	-287.6509	230.498	-1.248	0.212	-739.920	164.619
day[T.9-19]	-285.2692	228.621	-1.248	0.212	-733.856	163.318
day[T.9-2]	-294.8847	256.404	-1.150	0.250	-797.986	208.217
day[T.9-20]	-285.7735	226.717	-1.260	0.208	-730.625	159.078
day[T.9-21]	-286.8916	224.786	-1.276	0.202	-727.954	154.171
day[T.9-22]	-286.7113	222.829	-1.287	0.198	-723.934	150.511
day[T.9-23]	-286.9879	220.846	-1.299	0.194	-720.320	146.344
day[T.9-24]	-284.1894	218.838	-1.299	0.194	-713.581	145.202
day[T.9-25]	-283.9635	216.805	-1.310	0.191	-709.366	141.439
day[T.9-26]	-283.7007	214.749	-1.321	0.187	-705.068	137.666
day[T.9-27]	-282.4115	212.668	-1.328	0.184	-699.696	134.873
day[T.9-28]	-278.0487	210.565	-1.320	0.187	-691.206	135.109
day[T.9-29]	-278.8253	208.439	-1.338	0.181	-687.812	130.162
day[T.9-3]	-295.1125	255.031	-1.157	0.247	-795.518	205.293
day[T.9-30]	-278.0818	206.292	-1.348	0.178	-682.855	126.691
day[T.9-4]	-295.5013	253.622	-1.165	0.244	-793.143	202.141
day[T.9-5]	-293.8461	252.179	-1.165	0.244	-788.657	200.965
day[T.9-6]	-291.2269	250.702	-1.162	0.246	-783.140	200.686
day[T.9-7]	-289.5339	249.192	-1.162	0.246	-778.484	199.416
day[T.9-8]	-291.9620	247.649	-1.179	0.239	-777.884	193.960
day[T.9-9]	-292.4589	246.073	-1.189	0.235	-775.289	190.371
t	0.0096	0.001	6.770	0.000	0.007	0.012
I(t ** 2)	-4.292e-06	9.06e-07	-4.737	0.000	-6.07e-06	-2.51e-06
sint	-193.2119	87.946	-2.197	0.028	-365.774	-20.650
cost	-96.4567	131.035	-0.736	0.462	-353.566	160.653

Omnibus:	26.120	**Durbin-Watson:**	0.527
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	44.666
Skew:	-0.117	**Prob(JB):**	2.00e-10
Kurtosis:	3.824	**Cond. No.**	2.43e+10

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.43e+10. This might indicate that there are
strong multicollinearity or other numerical problems.

In [42]:
```
pred_df = pd.DataFrame()
t_pred = np.arange(len(df)+1, len(df)+366)
date_pred = pd.date_range(start = '01/01/2016', periods = 365, freq = 'd')
pred_df.index = date_pred
pred_df["t"] = t_pred
pred_df["sint"] = [math.sin((2*math.pi/365)*i) for i in pred_df["t"]]
pred_df["cost"] = [math.cos((2*math.pi/365)*i) for i in pred_df["t"]]
pred_df["day"] = [str(i.month) + "-" + str(i.day) for i in pred_df.index]
predictions = model.get_prediction(pred_df)
predictions = predictions.summary_frame(alpha = 0.05)
predictions.index = date_pred
```

In [43]:
```
plt.figure(figsize = (20, 5))
plt.plot(df.temp_avg)
plt.plot(model.fittedvalues)
plt.plot(predictions["mean"], color = "black")
plt.xlabel("Date")
plt.ylabel("Temperature")
plt.title("Daily Average Temperature in Seatle and 365-step Forecast")

plt.plot(predictions["mean_ci_lower"], color = "orange", linestyle = "--")
plt.plot(predictions["mean_ci_upper"], color = "orange", linestyle = "--")

plt.plot(predictions["obs_ci_lower"], color = "red", linestyle = "--")
plt.plot(predictions["obs_ci_upper"], color = "red", linestyle = "--")

plt.legend(["Observations", "Fit", "Predictions", "Lower CI", "Upper CI", "Lower PI", "Upper PI"])
plt.grid()
```



III. Conclusions and Future Work.

Conclusion: to achieve our goal to make 1-year-ahead forecast for average temprature in Seattle, we finally decide to use the model with formula: $Y_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \beta_3 sin(\frac{2\pi t}{365}) + \beta_4 cos(\frac{2\pi t}{365}) + \sum_{i=1}^{365} \alpha_i D_i$.

Compared to the other models we use in the 2.1 part, quadratic + periodic model has the best performance with highest adjust R-squared, best F-STAT, JB-STAT, t-stat of coefficients, lowest AIC and BIC, plus the condition that residuals are randomly scattering around 0 with constant variance. Our final model include the day dummies to make the model more comprehensive since we find out that there is clearly seasonal factors from the seasonal factor plot (increasing from Jan to Aug and decreasing from Aug to Dec). Our final model with coefficient being: $Y_t = 138.37 + 0.0096t - 0.0000043t^2 - 193.2119sin(\frac{2\pi t}{365}) - 96.4567cos(\frac{2\pi t}{365}) + \sum_{i=1}^{365} \alpha_i D_i$ with adjusted R-squared being 0.797 and all coefficients jointly statistically significant.

Future work: With significant spikes in acf plot and decreasing significance of autocorrelation between lags in pacf plot, we might consider use Moving Average to model the cycle in the residual, combining with our final model with trend and seasonality to make it more comprehensive. Moreover, we could calculate Mean Squared Error (MSE) and Mean Absolute Error (MAE) of our prediction to see how our prediction perform compared to the ground-truth values, and see how to improve our model from the resulting statistical metrics.

IV. References.

Ananth, R. (2021). Weather prediction [Data set]. Kaggle. https://www.kaggle.com/datasets/ananthr1/weather-prediction